Capítulo 2

Aprendizado Auto-Supervisionado Generativo e Contrastivo: Tendências e Desafios para Aplicações Dinâmicas em Redes

João Vitor Valle Silva, Guilherme Nunes Nasseh Barbosa,
Willian Tessaro Lunardi, Martin Andreoni,
Diogo Menezes Ferrazani Mattos

Abstract

This chapter addresses and contextualizes self-supervised learning as an alternative for dynamic network applications, where data labeling is a critical challenge due to the discrepancy between the traffic generation rate and the manual data labeling rate. Generative and contrastive self-supervised learning techniques come to the fore because they effectively improve network performance, expand the number of labeled samples, and enable recognition of similarities and differences between sample examples. Finally, self-supervised learning algorithms and their characteristics and network applications are presented, aiming to enable readers to understand this technique's principles, frameworks, and limitations.

Resumo

Este capítulo aborda e contextualiza o aprendizado auto-supervisionado como uma alternativa para aplicações dinâmicas de rede, em que a rotulagem de dados é um desafio crítico devido à discrepância entre a taxa de geração de tráfego e a taxa de rotulagem manual dos dados. São apresentadas técnicas de aprendizado auto-supervisionado generativo e contrastivo por serem eficazes para melhorar o desempenho da rede, expandindo o número de amostras rotuladas e reconhecendo semelhanças e diferenças entre exemplos de amostras. Por fim, são apresentados os algoritmos de aprendizado auto-supervisionado e suas características e aplicações em redes, visando capacitar os leitores a compreender os princípios, arcabouços e limitações dessa técnica.

2.1. Introdução

A computação ubíqua é parte fundamental do cotidiano das pessoas, principalmente na utilização de dispositivos inteligentes, como sensores, *wearables* e telefones inteligentes (*smartphones*), que compõem a Internet das Coisas. Para que a Internet das Coisas possa integrar cada vez mais dispositivos heterogêneos, é necessário desenvolver sistemas capazes de obter informações por meio da detecção e coleta de dados para controle e gerenciamento de múltiplas redes [Zafar et al., 2022]. Com bilhões de dispositivos conectados, a integração e interconexão entre eles representam um desafio para o gerenciamento de tráfego e otimização da rede. As redes móveis, tais como as redes de quinta geração (5G) e as previsões das redes de sexta geração (6G), são fundamentais para conectar dispositivos da Internet das Coisas. No entanto, um dos principais desafios para as redes 5G e além é suportar simultaneamente a implementação de Qualidade de Serviço (QoS) em um ambiente altamente heterogêneo, composto por diversos tipos de tráfego e aplicações adaptadas a diferentes requisitos, sob recursos de rede limitados e condições de rede dinâmicas [Zhang e Zhu, 2023]. O dinamismo existente nessas redes ainda é um desafio para aprimorar a implementação de uma arquitetura baseada em virtualização das funções de redes e redes definidas por software. Outro ponto importante é a adoção de uma arquitetura *Zero Trust*, que requer mecanismos para gerenciar eventos e informações de segurança (*Security Information and Event Management* - SIEM), além do armazenamento de registros de atividades (*logs*) de sistemas e redes [Stafford, 2020]. Para a análise dessas informações, é essencial que sejam utilizados algoritmos de aprendizado de máquina.

O aprendizado de máquina abrange vários paradigmas, cada um com características e aplicações distintas. Os quatro tipos principais de aprendizado de máquina são: (i) supervisionado, (ii) não-supervisionado, (iii) semi-supervisionado e (iv) auto-supervisionado. O aprendizado supervisionado envolve modelos de treinamento em conjuntos de dados rotulados, em que cada exemplo de treinamento está associado a um rótulo de saída. Em contraste, as tarefas de aprendizado não-supervisionado visam identificar padrões ou estruturas subjacentes em conjuntos de dados não rotulados. O aprendizado semi-supervisionado combina elementos de aprendizado supervisionado e não-supervisionado, utilizando uma pequena quantidade de dados rotulados juntamente com um conjunto maior de dados não rotulados. Por fim, o aprendizado auto-supervisionado é uma forma de aprendizado não-supervisionado em que o modelo gera rótulos a partir dos próprios dados, normalmente através de tarefas de pretexto, para aprender representações úteis. Cada tipo de aprendizado tem suas vantagens e desvantagens, tornando-os adequados para diferentes cenários e aplicações na área de aprendizado de máquina. A Tabela 2.1 apresenta as principais características de cada tipo de aprendizado de máquina.

Dentre as diferentes abordagens de aprendizado de máquina, as Redes Neurais Artificiais demonstram seu potencial diariamente em diversas aplicações, especialmente em tarefas de aprendizado supervisionado. O uso de redes neurais oferece a capacidade de processar grandes quantidades de dados, tornando tarefas como classificação de imagens, segmentação semântica, processamento de linguagem natural e aprendizado de grafos mais acessíveis e eficientes [Thisanke et al., 2023]. Embora os modelos de aprendizado supervisionado tenham se tornado ferramentas valiosas em diversas áreas, sua aplicação em problemas que envolvem dados não relacionais, como fluxos de rede, ainda apresenta

Tabela 2.1. Comparação entre aprendizado supervisionado, não-supervisionado, semi-supervisionado e auto-supervisionado

	Aprendizado Supervisionado	**Aprendizado Não Supervisionado**	**Aprendizado Semi-Supervisionado**	**Aprendizado Auto-Supervisionado**
Definição	Treina em dados rotulados	Treina em dados não rotulados	Treina com uma pequena quantidade de dados rotulados e uma grande quantidade de dados não rotulados	Gera rótulos a partir dos próprios dados e treina com esses rótulos gerados automaticamente
Requisitos de Dados	Grande quantidade de dados rotulados	Apenas dados não rotulados	Combinação de dados rotulados e não rotulados	Dados não rotulados com sinais de supervisão gerados
Treinamento	Aprende a mapear entradas para saídas com base nos rótulos fornecidos	Encontra padrões ou estruturas nos dados	Usa dados rotulados para guiar o aprendizado e captura padrões dos dados não rotulados	Resolve tarefas pretextuais criadas a partir dos dados para aprender representações úteis
Aplicações	Classificação, regressão, etc.	Agrupamento, redução de dimensionalidade, detecção de anomalias	Situações com escassez de dados rotulados (por exemplo, imagens médicas)	Pré-treinamento de modelos para tarefas de PLN e visão computacional
Exemplos	Detecção de spam em emails, previsão de preços de casas	Segmentação de clientes, PCA	Classificação de imagens com poucos exemplos rotulados	BERT para PLN, SimCLR para visão computacional
Vantagens	Previsões precisas com dados rotulados suficientes	Descobre padrões ocultos sem dados rotulados	Combina as vantagens do aprendizado supervisionado e não supervisionado	Aprende com dados não rotulados em grande escala, útil para pré-treinamento
Desvantagens	Requer grandes conjuntos de dados rotulados, que podem ser caros	Nem sempre encontra padrões úteis	Ainda requer alguns dados rotulados	Tarefas pretextuais complexas podem não estar sempre alinhadas com as tarefas posteriores

desafios. Esses modelos são altamente dependentes de conjuntos de dados rotulados, o que os torna suscetíveis a erros de generalização. A coleta de dados rotulados ou a rotulagem manual é difícil em diversos os aspectos. Treinar uma rede do zero é uma tarefa com alto custo computacional [Thisanke et al., 2023]. Por outro lado, os modelos auto-supervisionados têm se destacado como tendências de pesquisas recentes devido à sua eficiência em lidar com uma vasta quantidade de dados não rotulados e à sua alta capacidade de generalização. Ressalta-se que parte dos dados utilizados para o treinamento auto-supervisionado pode estar incompleta, ter sofrido transformações, como distorções ou traduções, ou estar parcialmente corrompida, como ocorre em algumas técnicas de aumento de dados (*data augmentation*). Nesses casos, o modelo aprende a recuperar a parte que sofreu o dano, todo o conjunto de dados ou simplesmente algumas características de interesse.

O aprendizado auto-supervisionado é visto como um ramo do aprendizado não-supervisionado, pois não há rótulo prévio associado às amostras. No entanto, o aprendizado não-supervisionado concentra-se em detectar padrões específicos nos dados, como agrupamentos, descoberta de comunidades ou detecção de anomalias, enquanto o aprendizado auto-supervisionado visa recuperar informações e, portanto, está alinhado ao paradigma de aprendizado supervisionado. O aprendizado auto-supervisionado pode ser categorizado de acordo com o treinamento realizado, sendo dividido em quatro categorias [Wu et al., 2023]:

- **Contrastivo**. A ideia central dos modelos contrastivos reside em tratar cada instância como uma classe distinta. As variantes da mesma instância são aproximadas no espaço de incorporação, enquanto as variantes de instâncias diferentes são separadas. Essas variantes são criadas através da aplicação de diferentes transformações nos dados originais;

- **Generativo**. Modelos generativos utilizam uma tarefa auto-supervisionada, em que o perfil original da instância é reconstruído a partir de versões corrompidas. O modelo é treinado para prever uma parte dos dados disponíveis, sendo as tarefas mais comuns a reconstrução da estrutura e das características;

- **Preditivo**. Embora os modelos preditivos e generativos possam parecer semelhantes, devido ao envolvimento de previsões em ambos, seus objetivos subjacentes divergem significativamente. Os métodos generativos direcionam seus esforços para a previsão de partes ausentes nos dados originais, o que pode ser interpretado como uma forma de autoprevisão. Em contrapartida, os métodos preditivos geram novas amostras ou rótulos a partir dos dados originais, visando auxiliar nas tarefas de pretexto;

- **Híbrido**. Combinar diversas tarefas auto-supervisionadas e integrá-las em um único modelo configura-se como uma estratégia viável. Essa abordagem híbrida geralmente demanda a utilização de múltiplos codificadores. Diferentes tarefas auto-supervisionadas podem ser executadas em paralelo ou colaborar entre si.

Os modelos generativos tiveram uma grande influência a partir do uso das Redes Adversariais Generativas (*Generative Adversarial Networks* - GANs). As GANs fornecem uma maneira de aprender representações profundas sem dados de treinamento extensivamente rotulados. Isso é possível derivando sinais de retropropagação por meio de um processo competitivo envolvendo um par de redes neurais: um gerador e um discriminador. O gerador é responsável por gerar dados, enquanto o discriminador é utilizado para distinguir entre os dados reais e gerados. Esse processo competitivo leva as duas redes a melhorar continuamente seu desempenho, resultando em representações profundas que capturam as características essenciais dos dados [Creswell et al., 2018]. No entanto, as Redes Adversárias Generativas (GANs) apresentam desafios particulares durante o treinamento dos modelos. Isso se deve à não convergência dos parâmetros, que oscilam de forma significativa. Além disso, o discriminador pode ser tão eficiente que impede a rede geradora de criar dados próximos da realidade, interrompendo o treinamento [Jaiswal et al., 2021].

O modelo contrastivo, por sua vez, configura-se como uma abordagem discriminativa. Seu objetivo é agrupar amostras semelhantes, aproximando-as entre si, enquanto afasta amostras distintas umas das outras [Jaiswal et al., 2021]. Nas abordagens discriminativas, as representações são aprendidas por meio da modelagem da distribuição condicional $p(y|x)$. Esse processo envolve duas etapas principais: a inferência, em que os valores das variáveis latentes $p(v|x)$ são inferidos a partir da entrada x, e a tomada de decisão, em que, com base nas variáveis latentes inferidas v, a decisão final sobre o rótulo y é tomada, representada por $p(y|v)$ [Le-Khac et al., 2020].

Este capítulo aborda os paradigmas de aprendizado auto-supervisionado, tanto generativo quanto contrastivo, e discute a aplicação desse tipo de aprendizado em atividades complexas nas redes de computadores, como Sistemas de Detecção de Intrusão e Sistemas Automatizados de Provisão de Qualidade de Serviço. Um dos principais desafios na aplicação do aprendizado auto-supervisionado ao tráfego de rede para detecção de anomalias reside na representação significativa dos dados. Mesmo com dados previamente rotulados, a detecção de anomalias continua sendo um grande desafio devido à constante criação de novos ataques, que podem se camuflar facilmente em grandes fluxos de rede. Por outro lado, a provisão de qualidade de serviço (QoS) enfrenta ambientes cada vez mais dinâmicos e heterogêneos, exigindo que o aprendizado auto-supervisionado seja parametrizado da maneira mais eficiente possível. O capítulo discute, então, as principais propostas de implementação de aprendizado auto-supervisionado, detalhando tarefas de pretexto e ressaltando a aplicabilidade das técnicas abordadas.

O restante do capítulo está organizado da seguinte forma. A Seção 2.2 apresenta tarefas de pretexto para a transformação dos dados e técnicas para a maximização da informação. O aprendizado auto-supervisionado generativo é discutido na Seção 2.3, enquanto o aprendizado auto-supervisionado contrastivo é abordado na Seção 2.4. A Seção 2.5 explora propostas de aprendizado auto-supervisionado baseadas em agrupamentos. Casos de uso de aprendizado auto-supervisionado em aplicações dinâmicas de redes de computadores são detalhados na Seção 2.6. As tendências de pesquisa, oportunidades e desafios são discutidos na Seção 2.7. A atividade prática proposta é descrita na Seção 2.8. Por fim, a Seção 2.9 conclui este capítulo.

2.2. Modelos de transformação e maximização da informação

Os modelos de transformação e a maximização da informação são importantes para compreensão do aprendizado auto-supervisionado. Os modelos de transformação referem-se a arquiteturas que podem gerar diversas transformações de dados, como rotações, translações ou mudanças de cores, aplicadas aos dados de entrada. Essas transformações são utilizadas como tarefas de pretexto no aprendizado auto-supervisionado, em que o modelo é treinado para prever a transformação aplicada aos dados de entrada. Tarefas de pretexto no aprendizado auto-supervisionado são tarefas artificialmente criadas que geram sinais de supervisão a partir de dados não rotulados, permitindo que modelos aprendam representações úteis dos dados. Exemplos comuns incluem a predição de rotação, na qual o modelo prevê a rotação aplicada a uma imagem; o preenchimento de partes faltantes, em que o modelo completa imagens ou textos com lacunas inseridas artificialmente; e a ordenação de blocos menores (*patches*) da imagem, em que o modelo reordena blocos embaralhados para a estrutura original. Essas tarefas ajudam o modelo a entender características significativas dos dados, que podem ser transferidas para outras tarefas supervisionadas, como classificação ou detecção de anomalias, facilitando o treinamento em grandes volumes de dados não rotulados.

A maximização da informação, por outro lado, é um princípio que orienta o processo de aprendizagem, incentivando o modelo a extrair o máximo possível de informações úteis dos dados de entrada. No aprendizado auto-supervisionado, a maximização da informação é alcançada através da concepção de tarefas de pretexto que exigem que o modelo capture características ou representações significativas dos dados de entrada. Ao

combinar modelos de transformação com maximização de informações, as abordagens de aprendizado auto-supervisionado podem efetivamente aproveitar dados não rotulados para aprender representações úteis sem exigir anotação manual.

Muitos métodos de aprendizado de representação auto-supervisionados fazem uso de transformações de imagem. Redes de Quebra-Cabeça e de Rotação aplicam transformações selecionadas a exemplos de imagem com o objetivo de prever a parametrização da transformação. Em contraste, outros métodos se concentram em aprender representações que são invariantes a certas transformações, o que pode levar a um fenômeno conhecido como colapso de representação. Este colapso descreve soluções triviais, como representações constantes, que atendem ao objetivo de invariância, mas oferecem pouco ou nenhum valor informativo para tarefas reais.

Para evitar esse colapso de representação, foram desenvolvidos os chamados métodos de maximização de informação, que formam uma classe de técnicas de representação focadas no conteúdo informativo das incorporações. Por exemplo, alguns desses métodos descorrelacionam explicitamente todos os elementos dos vetores de incorporação, evitando efetivamente o colapso e resultando em uma maximização indireta do conteúdo de informação. Esses métodos utilizam técnicas como a matriz de correlação cruzada normalizada de incorporações entre visualizações [Zbontar et al., 2021], a matriz de covariância para visualizações únicas [Bardes et al., 2022] e operações de embranquecimento para implementar essa abordagem [Ermolov et al., 2021]. A seguir, esses métodos são elencados e detalhados. Ressalta-se que grande parte dos trabalhos consideram imagens como dados de entrada. Sendo assim, neste capítulo os dados de entrada dos modelos são considerados também como imagens, a menos que seja explicitamente definido a natureza dos dados de entrada dos modelos considerados.

2.2.1. *Barlow Twins*

A ideia central por trás desse arcabouço é o princípio da redução de redundância. Esse princípio, introduzido por Barlow em 1961, afirma que a redução de redundância é crucial para a organização de mensagens sensoriais no cérebro. A estrutura por trás dessa técnica é ilustrada na Figura 2.1.

Para implementar esse princípio de redução de redundância, a abordagem *Barlow Twins* utiliza um conjunto de imagens X e cria duas visualizações $X(1) = t(X)$ e $X(2) = t(X)$ dessas imagens, em que $t \sim T$ é uma transformação que é amostrada aleatoriamente de T para cada imagem e cada visualização. Um codificador f_θ computa representações $Y(1) = f_\theta(X(1))$ e $Y(2) = f_\theta(X(2))$, que são alimentadas em um projetor g_ϕ para calcular projeções $Z(1) = [z(1)_1, ..., z(1)_n] = g_\phi(Y(1))$ e $Z(2) = [z(2)_1, ..., z(2)_n] = g_\phi(Y(2))$ para ambas as visualizações.

A ideia dos Barlow Twins é regularizar a matriz de correlação cruzada entre as projeções de ambas as visualizações. A matriz é calculada como

$$C = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{z(1)_i - \mu(1)}{\sigma(1)} \right) \left(\frac{z(2)_i - \mu(2)}{\sigma(2)} \right)^\top,$$

em que $\mu(j)$ e $\sigma(j)$ são a média e o desvio padrão sobre o conjunto de projeções da

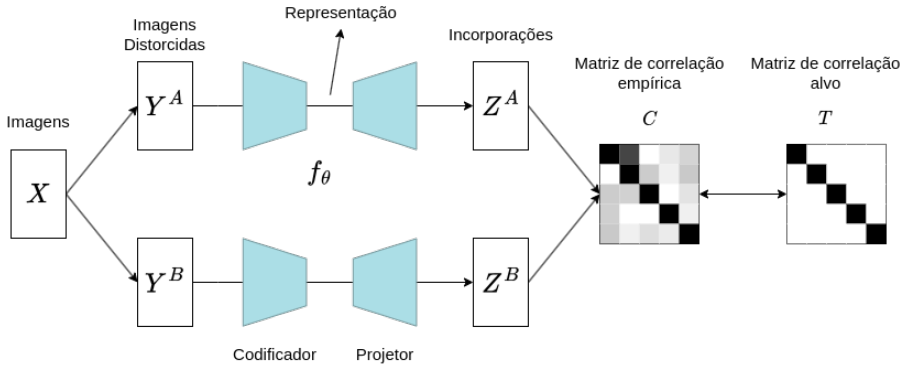

Figura 2.1. A técnica *Barlow Twins* busca tornar as representações obtidas a partir das redes neurais similares ao alimentá-las com versões distorcidas de um conjunto de dados. Isso reduz a redundância entre os componentes dos vetores de incorporação (*embedding*). É competitivo com outros métodos de autoaprendizagem, é simples conceitualmente, evita incorporações constantes triviais e é robusto ao tamanho do lote de treinamento. Adaptado de [Zbontar et al., 2021].

j-ésima visualização, calculadas como

$$\mu(j) = \frac{1}{n}\sum_{i=1}^{n} z(j)_i,$$

$$\sigma(j) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(z(j)_i - \mu(j))^2}.$$

A função de perda é então definida como

$$L_{BT}(\theta,\phi) = \sum_{k=1}^{d}(1-C[k,k])^2 + \lambda\sum_{k=1}^{d}\sum_{k'\neq k} C[k,k']^2,$$

em que d é o número de dimensões da projeção e $\lambda > 0$ é um hiperparâmetro. O parâmetro d promove invariância em relação às transformações aplicadas enquanto $\lambda > 0$ descorrelaciona as representações aprendidos, ou seja, reduz a redundância. Ao utilizar esta perda, o codificador f_θ é incentivado a prever representações que são descorrelacionadas, logo, que não são redundantes. Os *Barlow Twins* são treinados utilizando o otimizador LARS [You et al., 2017].

2.2.2. VICReg

VICReg (*Variance-Invariance-Covariance Regularization*) [Bardes et al., 2022] é um modelo auto-supervisionado de incorporação conjunta que se enquadra na categoria de métodos de maximização da informação. A proposta visa maximizar o acordo entre representações de diferentes visualizações de uma entrada, enquanto previne o colapso informacional usando dois termos de regularização adicionais. A Figura 2.2 representa a arquitetura básica da técnica VICReg, apresentando os seus principais elementos e operações.

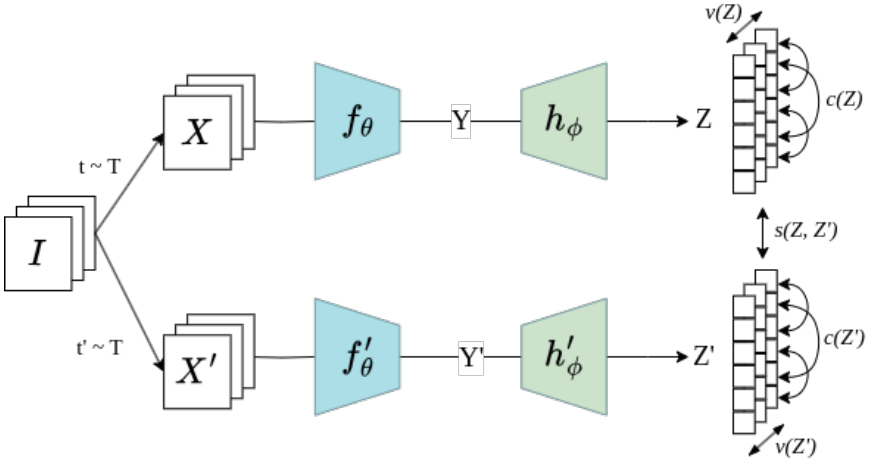

Figura 2.2. A técnica VICReg produz dois conjuntos de visualizações a partir de um lote de imagens, codifica essas visualizações em representações e, então, expande essas representações em incorporações. O objetivo é minimizar a distância entre incorporações da mesma imagem, manter a variância de cada variável de incorporação acima de um limite e atrair a covariância entre pares de variáveis de incorporação para zero. Adaptado de [Bardes et al., 2022].

Especificamente, a VICReg define termos de regularização para variância, invariância e covariância. Dado um lote de imagens X, duas visualizações $X(1) = t(X)$ e $X(2) = t(X)$ são definidas, onde $t \sim T$ é, novamente, amostrado aleatoriamente de T para cada imagem e cada visualização. Um codificador Siamese f_θ computa representações $Y(1) = f_\theta(X(1))$ e $Y(2) = f_\theta(X(2))$, que são alimentadas em um projetor Siamese g_ϕ para calcular projeções $Z(1) = [z(1)_1, ..., z(1)_n] = g_\phi(Y(1))$ e $Z(2) = [z(2)_1, ..., z(2)_n] = g_\phi(Y(2))$. Cada projeção possui d dimensões. Para cada visualização, a matriz de covariância das projeções é computada.

O termo de variância visa manter o desvio padrão de cada elemento do incorporação acima de uma margem b. Praticamente, isso impede que os vetores de incorporação sejam os mesmos em todo o lote e é um dos dois mecanismos que visam prevenir o colapso. Pode ser implementado usando uma perda de bisel. O termo de covariância descorrelaciona os elementos dos vetores de incorporação para visualizações únicas a fim de reduzir a redundância e evitar o colapso. Isso é alcançado minimizando os elementos fora da diagonal ao quadrado da matriz de covariância em direção a 0. Por fim, o termo de invariância é usado para maximizar o acordo entre duas projeções da mesma imagem, induzindo invariância às transformações aplicadas a x_i. Para isso, é calculado o erro quadrático médio entre as projeções. Em geral, a perda da VICReg pode ser definida como a soma ponderada de todas as três regularizações para as visualizações fornecidas, onde os parâmetros λ_V, λ_I e λ_C balanceiam as perdas individuais.

2.2.3. Técnica de Embranquecimento para Representações

Embranquecimento é uma transformação linear aplicada a um conjunto de dados, tornando-os descorrelacionados e com variância unitária, ou seja, a matriz de covariância torna-se a matriz identidade. O método de Embranquecimento com Perda Por Mínimos Quadrados (*Whitening Mean Squared Error* - W-MSE) [Ermolov et al., 2021] aplica essa ideia às incorporações de imagens para prevenir o colapso da representação.

Dado um lote de imagens X, transformações aleatórias são aplicadas para obter m visualizações $X(j)$ para todo $j \in \{1,...,m\}$. Um codificador Siamese f_θ mapeia as visualizações para representações $Y(j) = f_\theta(X(j))$, que são então alimentadas em um projetor Siamese g_ϕ para calcular projeções $Z(j) = [z(j)_1,...,z(j)_n] = g_\phi(Y(j))$. Todas as projeções são então concatenadas em uma única matriz $Z = [z(1)_1,...,z(1)_n,...,z(m)_1,...,z(m)_n]$. Esta matriz é branqueada para obter $Z^\sim$ removendo a média e descorrelacionando-a usando a decomposição de Cholesky da matriz de covariância inversa. Para treinar os modelos, o erro quadrático normalizado entre todos os pares de projeções branqueadas é minimizado. A função de perda é definida como

$$L_{WMSE}(\theta,\phi) = \frac{1}{n}\sum_{i=1}^{n}\frac{2}{m(m-1)}\sum_{j=1}^{m}\sum_{k=j+1}^{m} d_{nse}(\tilde{z}_i^{(j)},\tilde{z}_i^{(k)}).$$

O passo de branqueamento é essencial para evitar o colapso das representações. O objetivo é maximizar a similaridade entre todos os pares aumentados, enquanto também previne o colapso da representação ao impor covariância unitária nas projeções.

2.2.4. Modelo *Transformer* de redes neurais artificiais

O modelo *Transformer* é uma arquitetura de rede neural avançado no campo do processamento de linguagem natural (*Natural Language Processing* - NLP) [Vaswani et al., 2017]. Embora, o modelo *Transformer* não seja considerado um modelo de aprendizado auto-supervisionado, esse modelo apresenta algumas características que são importantes para a definição do aprendizado auto-supervisionado. Diferentemente dos modelos tradicionais, que empregam camadas recorrentes, o *Transformer* utiliza um mecanismo chamado **atenção** para capturar dependências de longo alcance entre os elementos de uma sequência [Torbarina et al., 2024]. Essa abordagem permite uma maior paralelização durante o treinamento e resulta em uma melhor qualidade de tradução em tarefas de NLP, como a tradução de idiomas. O modelo de atenção é crucial para o *Transformer*, permitindo que cada elemento em uma sequência seja comparado com todos os outros para calcular sua importância relativa, possibilitando a aprendizagem de dependências de longo alcance de forma eficaz.

Além da autoatenção dentro do codificador e do decodificador, o *Transformer* também utiliza atenção entre o codificador e o decodificador [Vaswani et al., 2017]. Essa atenção cruzada permite que o modelo se concentre em partes relevantes da entrada durante a geração da saída, facilitando a captura de dependências entre a entrada e a saída. A arquitetura do *Transformer* é composta por várias camadas empilhadas de blocos de atenção e redes *feedforward*, tanto no codificador quanto no decodificador [Vaswani et al., 2017].

O *EsVit* (*Efficient Self-Supervised Vision Transformers*) propõe uma abordagem inovadora para a aplicação de *Transformers* em tarefas visuais [Li et al., 2021]. Essa arquitetura professor-aluno substitui os *Transformers* Visuais por transformadores multi-estágio, mesclando blocos menores (*patches*) de imagem em cada camada para reduzir o processamento [Caron et al., 2021]. No entanto, para mitigar a perda de correspondências locais importantes, o *EsVit* introduz uma perda de correspondência de região adicional [Li et al., 2021]. Isso é feito através de uma extensão da função de perda para identificar

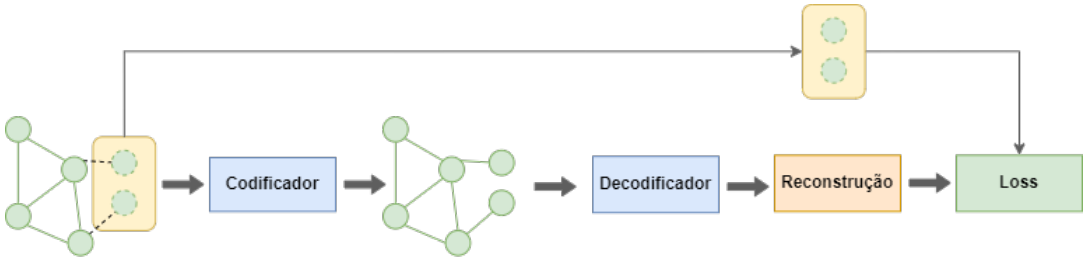

Figura 2.3. Estrutura básica de um modelo auto-supervisionado generativo. O método auto-supervisionado generativo concentra-se nas informações codificadas no grafo, frequentemente baseadas em tarefas de reconstrução de subgrafos ou de elementos do grafo. Desse modo, os atributos e as estruturas dos dados do grafo são utilizados como sinais de auto-supervisão, guiando o aprendizado do modelo sem a necessidade de rótulos externos. Adaptado de [Wu et al., 2023].

características em nível de região, combinando uma perda de nível de visualização e uma perda de nível de região [Li et al., 2021]. Essa abordagem mostra melhorias significativas na captura de correspondências, especialmente em arquiteturas multiestágio, superando limitações anteriores.

2.3. Aprendizado auto-supervisionado generativo

O aprendizado auto-supervisionado generativo se caracteriza pela geração de informações a partir de dados brutos, sem a necessidade de rótulos externos. Através de um decodificador, o modelo reconstrói os dados de entrada, extraindo conhecimento da própria estrutura da informação. Em outras palavras, o modelo busca prever e recriar os dados originais de forma autônoma, sem depender de tarefas supervisionadas externas, conforme ilustrado na Figura 2.3. Existem duas abordagens principais no aprendizado auto-supervisionado generativo: **codificadores automáticos** e **modelos auto-regressivos**. Os codificadores automáticos tem por objetivo reconstruir os dados de entrada de uma só vez, enquanto os modelos auto-regressivos fazem isso de forma iterativa, prevendo um elemento dos dados de cada vez com base nos elementos anteriores.

2.3.1. Modelos auto-regressivos

Os **Modelos auto-regressivos (AR)** são ferramentas estatísticas que descrevem a relação entre variáveis sequenciais, sendo amplamente utilizados na análise de séries temporais. São frequentemente utilizados em cenários estacionários, cuja a média é constante ao longo do tempo [Barbosa et al., 2021a]. A distribuição conjunta pode ser fatorada como um produto de condicionais, em que a probabilidade de cada variável depende das variáveis anteriores [Liu et al., 2023], de acordo com:

$$\max_{\theta} p_{\theta}(x) = \sum_{t=1}^{T} \log p_{\theta}(x_t | x_{1:t-1}).$$

em que $p_{\theta}(x)$ representa a distribuição de probabilidade conjunta de todas as variáveis na série temporal x, onde θ denota os parâmetros do modelo.

Na modelagem de linguagem auto-regressiva em processamento de linguagem natural (*Natural Language Processing* - NLP), como em GPT-4 [Lee et al., 2023], o ob-

jetivo é maximizar a verossimilhança sob a fatoração auto-regressiva para representações unificadas em diferentes áreas, desde processamento de linguagem natural até visão computacional e geração de grafos.

Os modelos autorregressivos também têm sido empregados em visão computacional, como no PixelRNN [Oord et al., 2016b] e PixelCNN [Van den Oord et al., 2016]. Com base no PixelCNN, foi proposto o WaveNet [Oord et al., 2016a], um modelo generativo para áudio bruto. Para lidar com dependências temporais de longo alcance, os autores desenvolveram convoluções causais dilatadas para melhorar o campo receptivo. Além disso, blocos residuais com portas e conexões de salto são empregados para aumentar a expressividade.

A adaptação de domínio é uma técnica de aprendizado de máquina que visa aprimorar o desempenho de um modelo em um domínio de destino, utilizando dados de um domínio de origem relacionado, porém diferente. O objetivo é permitir que um modelo treinado em um conjunto de dados (domínio de origem) generalize seu aprendizado para um conjunto de dados diferente (domínio de destino). A adaptação de domínio pode ser supervisionada, semi-supervisionada ou não supervisionada, dependendo da disponibilidade de rótulos nos conjuntos de dados de origem e destino. Em problemas que envolvem séries temporais, para resolver a Adaptação de Domínio Não Supervisionada (*Unsupervised Domain Adaptation* - UDA), surgem desafios adicionais. Por exemplo, a maioria das soluções existentes é desenvolvida especificamente para dados visuais, e muitas abordagens de adaptação de domínio dependem do pré-treinamento em um grande banco de dados visuais, tal como o ImageNet, como inicialização do modelo. Nesse contexto, [Ragab et al., 2022] propõe uma nova estrutura de Adaptação de Domínio AutoRegressiva Auto-supervisionada (*Self-Supervised AutoRegressive Domain Adaptation* - SLARDA) para aprimorar o desempenho da UDA em séries temporais.

O SLARDA emprega aprendizado autoregressivo por meio de um discriminador autoregressivo que considera a dependência temporal entre as características de séries temporais de origem e destino durante o alinhamento de domínio. O discriminador autoregressivo consiste em uma rede autoregressiva que codifica as dependências temporais entre as características de ambos os domínios em representações vetoriais. Essa abordagem permite capturar a dinâmica temporal dos dados de séries temporais durante o processo de alinhamento de domínio, o que é crucial para obter uma melhor adaptação entre os domínios.

Ao considerar a dependência temporal, o discriminador autoregressivo pode discernir melhor entre as características de séries temporais de origem e destino, evitando ser enganado e alcançando um estado de alinhamento mais satisfatório. Isso contrasta com abordagens anteriores que ignoram a dimensão temporal ao discriminar entre as características de séries temporais de diferentes domínios, resultando em um desempenho limitado para o alinhamento de domínio. Portanto, ao incorporar o aprendizado autoregressivo no SLARDA, a abordagem consegue capturar a dependência temporal das características de séries temporais, melhorando significativamente a capacidade de alinhamento de domínio e, consequentemente, o desempenho geral da adaptação de domínio.

Os modelos autorregressivos também podem ser aplicados a problemas de domínio de grafos, como a geração de grafos. O trabalho de [You et al., 2018b] propõe o

GraphRNN para gerar grafos realistas com modelos autorregressivos profundos. Eles decompõem o processo de geração de grafos em uma sequência de geração de nós e arestas condicionadas ao grafo gerado até o momento. O objetivo do GraphRNN é definido como a verossimilhança das sequências de geração de grafos observadas. O GraphRNN pode ser visto como um modelo hierárquico, em que uma rede neural recorrente (*Recurrent Neural Network* - RNN) de nível de grafo mantém o estado do grafo e gera novos nós, enquanto uma RNN de nível de aresta gera novas arestas com base no estado atual do grafo. Como consequência, outras propostas também se baseiam em abordagens autorregressivas, tais como MRNN [Popova et al., 2019] e GCPN [You et al., 2018a].

O GPT-GNN [Li et al., 2024b] propõe uma estrutura autoregressiva para realizar a reconstrução de nós e arestas em um grafo dado de forma iterativa. Dado um grafo $g_t = (A_t, X_t)$ com seus nós e arestas mascarados aleatoriamente na iteração t, o GPT-GNN gera um nó mascarado X_i e suas arestas conectadas E_i para obter um grafo atualizado $g_{t+1} = (A_{t+1}, X_{t+1})$ e otimiza a probabilidade de geração de nós e arestas na próxima iteração $t+1$, com o objetivo de aprendizado definido como

$$
\begin{aligned}
& p_\theta(X_{t+1}, A_{t+1} | X_t, A_t) \\
& = \sum_o p_\theta(X_i, E_{\neg o_i} | E_{oi}, X_t, A_t) \cdot p_\theta(E_{oi} | X_t, A_t) \\
& = E_o p_\theta(X_{t+1} | E_{oi}, X_t, A_t) \cdot p_\theta(E_{\neg o_i} | E_{oi}, X_{t+1}, A_t),
\end{aligned}
$$

em que o é uma variável que denota o vetor de índice de todas as arestas dentro de E_t na iteração t. Assim, E_{oi} denota as arestas observadas na iteração t, e $E_{\neg o_i}$ denota as arestas mascaradas na iteração $t+1$. Finalmente, o processo de geração de grafo é fatorado em uma etapa de geração de atributo de nó $p_\theta(X_{t+1} | E_{oi}, X_t, A_t)$ e uma etapa de geração de aresta $p_\theta(E_{\neg o_i} | E_{oi}, X_{t+1}, A_t)$.

2.3.2. Modelos baseados em auto-codificadores

Modelos baseados em codificador automático (*autoencoders* - AE) são uma classe popular de modelos utilizados em aprendizado de máquina, especialmente em tarefas de reconstrução e geração de dados. Em diversas situações, atua como uma etapa preparatória para outras redes neurais, diminuindo a complexidade dos dados e eliminando possíveis ruídos [Barbosa et al., 2021b]. Essa otimização beneficia o aprendizado de algoritmos supervisionados. A estrutura do codificador automático é composta por três camadas distintas: entrada, oculta e saída. A **camada oculta** atua como um codificador, compactando as informações, enquanto a **camada de saída** funciona como o decodificador, reconstruindo os dados originais com base na representação latente [Bochie et al., 2020]. Durante a fase de codificação, o modelo reduz a dimensionalidade dos dados de entrada, capturando suas principais características em uma representação latente de menor dimensão. Em seguida, na fase de decodificação, o modelo tenta reconstruir os dados originais a partir da representação latente. Esses modelos têm sido amplamente explorados em diferentes variações, como o *autoencoder denoising*, que é treinado para reconstruir dados a partir de versões corrompidas ou ruidosas, e o *autoencoder* variacional (*Variational Autoecoder* - VAE), que introduz uma abordagem probabilística na geração de dados, permitindo a aprendizagem de representações latentes mais ricas e estruturadas.

Modelo de previsão de contexto

Cada variação dos codificadores automáticos tem suas próprias características e aplicações específicas. Por exemplo, o modelo de previsão de contexto (*Context Prediction Model* - CPM) é útil em tarefas em que a contextualização das informações é importante, enquanto o VAE é valioso quando se deseja aprender representações mais significativas dos dados, especialmente em contextos em que a incerteza é relevante. A escolha da arquitetura, da função de perda e da técnica de treinamento é fundamental para o desempenho desses modelos em diferentes contextos. Esses modelos são frequentemente aplicados em uma variedade de domínios, incluindo processamento de linguagem natural, visão computacional e geração de dados, demonstrando sua versatilidade e eficácia em várias aplicações de aprendizado de máquina.

A ideia do Modelo de Predição de Contexto (CPM) é prever informações contextuais com base nas entradas. Em Processamento de Linguagem Natural (PLN), quando se trata de aprendizado auto-supervisionado *word embedding*, *CBOW* e *Skip-Gram* [Mikolov et al., 2013, de Oliveira et al., 2021] são trabalhos pioneiros. O objetivo do CBOW é prever os *tokens* de entrada com base nos *tokens* de contexto. Em contraste, o objetivo do Skip-Gram é prever os *tokens* de contexto com base nos *tokens* de entrada. Normalmente, a amostragem negativa é empregada para garantir eficiência computacional e escalabilidade.

Inspirados pelo progresso dos modelos de *word embedding* em PLN, muitos modelos de incorporação de rede são propostos com base em um objetivo semelhante de predição de contexto. O Deepwalk [Perozzi et al., 2014] amostra caminhadas aleatórias truncadas para aprender a incorporação latente de nós com base no modelo Skip-Gram. O modelo trata caminhadas aleatórias como o equivalente a frases. No entanto, outra abordagem de incorporação de rede, o LINE [Tang et al., 2015], visa gerar vizinhos em vez de nós em um caminho com base nos nós atuais:

$$O = -\sum_{(i,j)\in E} w_{ij} \log p(v_j|v_i),$$

em que E denota o conjunto de arestas, v denota o nó, w_{ij} representa o peso da aresta (v_i, v_j). O LINE também usa amostragem negativa para amostrar múltiplas arestas negativas a fim de aproximar o objetivo.

Modelo de redução de ruído

Uma das variações de modelos que utilizam codificadores automáticos é o *Modelo de Redução de Ruído (Denoising Autoencoder)*. Tradicionalmente, o codificador automático não é capaz de obter características relevantes em ambientes heterogêneos. Para contornar esse problema, o modelo de redução de ruído é treinado para reconstruir os dados de entrada após aplicação de ruídos [Abusitta et al., 2023].

A intuição é que a representação deve ser robusta à introdução de ruído. O modelo de linguagem mascarada (*Masked Language Model* - MLM), uma das arquiteturas mais bem-sucedidas no processamento de linguagem natural, pode ser considerado como um modelo de codificador automático de remoção de ruído. Para modelar sequências de texto, o modelo de linguagem mascarada (MLM) oculta aleatoriamente alguns dos

tokens da entrada e então os prevê com base em suas informações de contexto. BERT [Devlin et al., 2018] é o trabalho mais representativo nesse campo. Especificamente, no BERT, um *token* único $[MASK]$ é introduzido no processo de treinamento para mascarar alguns *tokens*. No entanto, uma limitação desse método é que não há *tokens* de entrada $[MASK]$ para tarefas posteriores. Para mitigar isso, os autores nem sempre substituem os *tokens* previstos por $[MASK]$ durante o treinamento. Em vez disso, eles os substituem por palavras originais ou palavras aleatórias com uma pequena probabilidade.

Outra abordagem promissora na aprendizagem de representação envolve a combinação de modelos de condrificadores automáticos e baseados em fluxo. Modelos baseados em fluxo mapeiam uma distribuição de base para a distribuição de interesse, gerando amostras de alta qualidade e calculando a densidade de probabilidade exata. O modelo GraphAF [Liu et al., 2019a] é um exemplo dessa combinação usado no contexto de geração de moléculas. Além disso, incorpora conhecimento detalhado do domínio na definição da recompensa, como a verificação de valência.

A técnica de “dequantização"também é aplicada para converter dados discretos em contínuos, sendo útil em tarefas de processamento de imagem. Essa técnica é particularmente útil em tarefas de processamento de imagem, em que os dados são frequentemente discretos. Essas abordagens oferecem vantagens significativas na geração de amostras de alta qualidade e no cálculo preciso de densidades de probabilidade em diversos domínios de aplicação. A combinação de modelos *autoencoders* e baseados em fluxo é uma área ativa de pesquisa com potencial aplicação em diversas áreas, como o processamento de tráfego de redes.

Modelos híbridos

Modelos híbridos combinam modelos auto-regressivos (AR) e *autoencoders* (AE) e são uma abordagem promissora para obter representações latentes mais ricas e estruturadas dos dados. Essa abordagem é útil em tarefas de processamento de linguagem natural. Além disso, a combinação de modelos AR e AE pode ser estendida para outras áreas de pesquisa, como processamento de imagem, processamento de sinais e processamento de tráfego de redes.

Alguns pesquisadores propõem combinar as vantagens dos modelos autoregressivos (AR) e dos *autoencoders* (AE). O modelo MADE (*Masked Autoencoder for Distribution Estimation*) [Germain et al., 2015] faz uma modificação simples no *autoencoder*, mascarando os parâmetros para respeitar as restrições autoregressivas. Especificamente, enquanto os neurônios entre camadas adjacentes são totalmente conectados no *autoencoder* original, no MADE, algumas conexões são mascaradas para garantir que cada dimensão de entrada seja reconstruída apenas a partir de suas próprias dimensões. MADE pode ser facilmente paralelizado em computações condicionais, permitindo estimativas diretas e econômicas de probabilidades conjuntas em alta dimensão.

No processamento de linguagem natural (PLN), o Modelo de Linguagem por Permutação (*Permutation Language Model* - PLM) é um modelo representativo que combina as vantagens dos modelos autoregressivos e dos *autoencoders*. O XLNet, que introduz o PLM, é um método de pré-treinamento autoregressivo generalizado [Yang et al., 2019]. O XLNet permite aprender contextos bidirecionais maximizando a probabilidade esperada

sobre todas as permutações da ordem de fatoração.

Para formalizar a ideia, seja Z_T o conjunto de todas as permutações possíveis da sequência de índices de comprimento T $[1,2,...,T]$, o objetivo do PLM pode ser expresso como:

$$\max_{\theta} \mathbb{E}_{z \sim Z_T} \left[\sum_{t=1}^{T} \log p_\theta (x_{z_t} | x_{z<t}) \right].$$

Para cada sequência de texto, diferentes ordens de fatoração são amostradas, permitindo que cada *token* veja sua informação contextual de ambos os lados. Com base na ordem permutada, o XLNet também realiza a reparametrização com posições para que o modelo saiba qual posição precisa prever. Então, uma atenção especial em dois fluxos é introduzida para a previsão consciente do alvo.

Além disso, diferentemente do BERT, inspirado pelos avanços recentes no modelo AR, o XLNet integra o mecanismo de recorrência de segmentos e o esquema de codificação relativa do Transformer-XL [Dai et al., 2019] no pré-treinamento, o que permite modelar melhor a dependência de longo alcance em comparação com o modelo Transformer [Torbarina et al., 2024].

DINO - *Self-Distillation With No Labels*

O DINO [Caron et al., 2018] define uma rede aluno e uma rede professor. O aluno consiste em um codificador f_θ e um projetor g_ϕ com parâmetros θ e ϕ. O codificador é implementado como um *transformer* visual e o projetor como uma rede perceptron de múltiplas camadas (*Multi-Layer Perceptron* - MLP). O professor consiste em um codificador $f_{\bar{\theta}}$ e um projetor $g_{\bar{\phi}}$ com a mesma arquitetura do aluno, mas um conjunto separado de parâmetros $\bar{\theta}$ e $\bar{\phi}$. A arquitetura do Modelo é apresentada na Figura 2.4.

DINO usa uma estratégia *multi-crop* primeiro para criar um lote de m visualizações $X_i = [x_{(1)i}, \ldots, x_{(m)i}]$ de uma imagem x_i. Cada visualização é uma recorte aleatório de x_i seguido por mais transformações. A maioria dos recortes cobre uma pequena região da imagem, mas alguns recortes são de alta resolução, referidos como visualizações locais e globais, respectivamente. Seja M_i o conjunto de índices das visualizações globais. A ideia é que a rede aluno tenha acesso a todas as visualizações, enquanto a rede professor tenha acesso apenas às visualizações globais, criando correspondências “do local para o global" [Caron et al., 2021].

O aluno calcula representações $y_{(j)i} = f_\theta(x_{(j)i})$ e projeções $z_{(j)i} = g_\phi(y_{(j)i})$ para cada visualização. O professor calcula projeções alvo $\bar{z}_{(j)i} = g_{\bar{\phi}}(f_{\bar{\theta}}(x_{(j)i}))$ para as visualizações globais $j \in M_i$.

Para prevenir o colapso, os autores identificam experimentalmente duas formas distintas em que ele pode ocorrer, a distribuição de probabilidade calculada pode ser uniforme ou uma única dimensão pode dominar, independentemente da entrada. Isso motiva a adoção de duas contramedidas específicas:

- Para evitar o colapso para uma distribuição uniforme, a distribuição alvo do professor é ajustada definindo o parâmetro de temperatura τ para um valor pequeno.

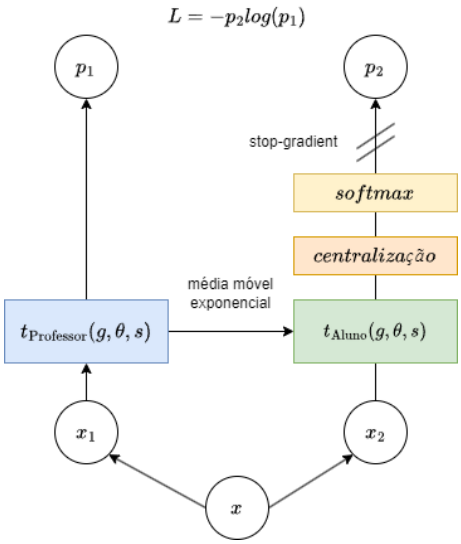

Figura 2.4. Arquitetura do Modelo DINO utiliza um par de imagens transformadas aleatoriamente para treinar redes de aluno e professor com a mesma arquitetura, mas diferentes parâmetros. As saídas das redes são normalizadas e comparadas usando uma perda de entropia cruzada. Um operador *stop-gradient* é aplicado ao professor para propagar gradientes apenas para o aluno. Os parâmetros do professor são atualizados usando uma média móvel exponencial dos parâmetros do aluno. Adaptado de [Caron et al., 2018].

- Para evitar que uma dimensão domine, a saída do professor é centralizada para torná-la mais uniforme.

Isso é realizado adicionando um vetor de centralização c como um viés ao professor, que é calculado com uma média móvel exponencial:

$$c \leftarrow \beta c + (1-\beta)\bar{z}, \quad \text{onde } \beta \in [0,1]$$

é um hiperparâmetro de decaimento que determina em que medida o vetor de centralização é atualizado, e

$$\bar{z} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|M_i|}\sum_{j\in M_i} z_{(j)i}$$

é a média de todas as projeções do professor no lote atual.

Para realizar o aprendizado de características invariantes por meio de rótulos suaves, o método DINO formula a tarefa de prever as projeções alvo como uma tarefa de destilação de conhecimento. As projeções do professor e do aluno são convertidas em distribuições de probabilidade, aplicando a função *softmax* sobre todos os componentes. Assim, a perda de entropia cruzada pode ser aplicada, em que o professor gera rótulos suaves para o aluno. A função de perda total associa cada visualização do aluno com cada visualização global do professor, dado por

$$L_{\text{DINO}}^{\theta,\phi} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j\in M_i}\sum_{k\neq j}\text{dce}\left(\text{softmax}_\tau(\bar{z}_{(j)i} - c), \text{softmax}_\rho(z_{(k)i})\right),$$

em que $\tau, \rho > 0$ são hiperparâmetros que controlam a temperatura das distribuições para o professor e o aluno, respectivamente. No geral, as atualizações de parâmetros são muito semelhantes às do BYOL, uma vez que a rede aluno é atualizada minimizando a perda $L_{\text{DINO}}^{\theta,\phi}$ usando o otimizador AdamW, e a rede professor é atualizada por uma média móvel exponencial do aluno, ou seja,

$$\bar{\theta} \leftarrow \alpha\bar{\theta} + (1-\alpha)\theta, \quad \bar{\phi} \leftarrow \alpha\bar{\phi} + (1-\alpha)\phi,$$

em que $\alpha \in [0,1]$ controla a taxa na qual os pesos da rede professor são atualizados com os pesos da rede aluno.

2.4. Aprendizado auto-supervisionado contrastivo

Do ponto de vista estatístico, os modelos de aprendizado de máquina são categorizados em modelos generativos e discriminativos. Dada a distribuição conjunta $P(X,Y)$ da entrada X e do alvo Y, o modelo generativo calcula $P(X|Y=y)$ enquanto o modelo discriminativo visa modelar $P(Y|X=x)$.

2.4.1. Aprendizado contrastivo de representação

Embora não seja possível avaliar $P(X)$ ou $P(X|Y)$ diretamente, pode-se usar amostras dessas distribuições, permitindo usar técnicas como Estimação por Contraste de Ruído [Oord et al., 2018, Gutmann e Hyvärinen, 2010], que se baseia na comparação do valor alvo com valores negativos amostrados aleatoriamente. A técnica de Estimação por Contraste de Ruído (*Noise-Contrastive Estimation* - NCE) é um método de estimação de parâmetros em modelos estatísticos parametrizados, que foi proposto como uma abordagem eficiente para lidar com modelos estatísticos não normalizados. NCE introduz um ruído contrastivo no processo de estimação, permitindo que a normalização do modelo seja tratada como um problema de otimização. A ideia geral do modelo de aprendizado contrastivo é apresentada na Figura 2.5.

Estimação de Ruído Contrastivo - *Noise Contrastive Estimation* - NCE

A ideia da Estimação de Ruído Contrastivo é formular a tarefa de aprendizado de representação contrastiva como um problema de classificação supervisionada. Uma suposição do NCE é que os negativos são independentes do âncora, ou seja, $p_{\text{neg}}(x^-|x^*) = p_{\text{neg}}(x^-)$. Nesse contexto, os negativos são frequentemente chamados de ruído. Existem duas abordagens amplamente utilizadas, o NCE original e o InfoNCE [Oord et al., 2018]. De forma geral, o NCE realiza classificação binária para decidir se uma amostra individual é positiva ou negativa, enquanto o InfoNCE realiza classificação multi-classe em um conjunto de amostras para decidir qual é a positiva.

A função objetivo da Estimação de Ruído Contrastivo (NCE) é dada por:

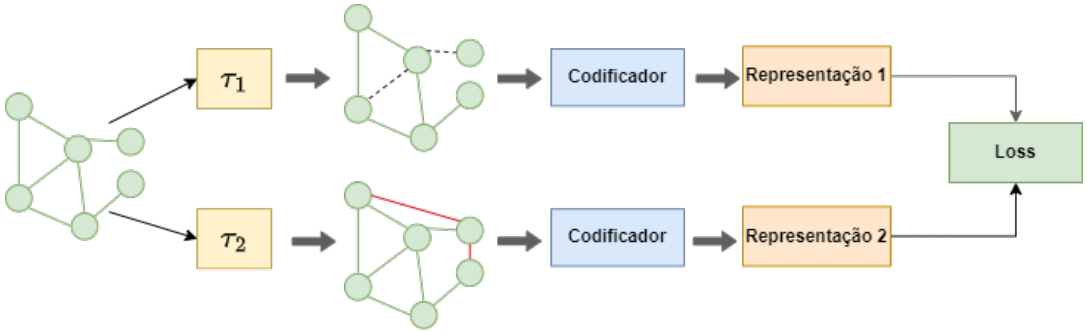

Figura 2.5. Estrutura básica de um modelo auto-supervisionado contrastivo. O método contrastivo compara as informações geradas por diferentes funções T1 e T2. A informação sobre as diferenças e semelhanças entre pares de dados (inter-dados) é usada como sinais de auto-supervisão. Adaptado de [Wu et al., 2023].

$$L = \mathbb{E}_{x,x^+,x^-}\left[-\log\left(\frac{e^{f(x)/T}}{e^{f(x)^T f(x^+)}} + e^{f(x)^T f(x^-)}\right)\right],$$

em que x^+ é similar a x, x^- é dissimilar a x e f é um codificador (função de representação).

InfoNCE

Para cada âncora x^*, o InfoNCE gera uma amostra positiva a partir de $p_{\text{pos}}(x^+|x^*)$ e $n-1$ amostras negativas de $p_{\text{neg}}(x^-)$. Seja $X = [x_1,\ldots,x_n]$ o conjunto dessas amostras, em que x_c é a positiva com índice $c \in \{1,\ldots,n\}$. No contexto do aprendizado de representação, calculam-se representações adicionais usando um codificador f_θ e se obtém o conjunto $Y = [y_1,\ldots,y_n]$. O InfoNCE agora define uma tarefa de classificação supervisionada, em que a entrada é (y^*,Y) e o rótulo de classe é o índice do positivo c. Um classificador $p_\psi(c|Y,y^*)$ com parâmetros ψ é treinado para combinar a verdadeira distribuição de dados dos rótulos $p_{\text{data}}(c|Y,y^*)$. Um objetivo comum de aprendizado supervisionado é minimizar a entropia cruzada entre a distribuição de dados e a distribuição do modelo, ou seja,

$$\min_{\psi,\theta} \mathbb{E}_{Y,y^*}[H(p_{\text{data}}(c|Y,y^*), p_\psi(c|Y,y^*))] = \min_{\psi,\theta} \mathbb{E}_{Y,y^*}[H(p_{\text{data}}(c|Y,y^*), p_\psi(c|Y,y^*))].$$

Este é um problema de previsão anti-causal, em que a causa subjacente (rótulo) é prevista a partir de seu efeito. No InfoNCE, é conhecido o mecanismo subjacente, já que os rótulos são gerados artificialmente, então pode-se derivar o classificador ideal usando o teorema de Bayes.

Primeiro, estima-se a distribuição de dados de um conjunto Y dado um rótulo e uma âncora, ou seja,

$$p_{\text{data}}(Y|c,y^*) = \prod_{i=1}^{n} p_{\text{data}}(y_i|c,y^*) = \prod_{i=1}^{n} \begin{cases} p_{\text{pos}}(y_i|y^*), & \text{se } i = c, \\ p_{\text{neg}}(y_i), & \text{se } i \neq c, \end{cases}$$

$$= p_{\text{pos}}(y_c|y^*) \prod_{i \neq c} p_{\text{neg}}(y_i) = p_{\text{pos}}(y_c|y^*) \cdot \left(\prod_{i=1}^{n} p_{\text{neg}}(y_i) \right),$$

em que se assume independência condicional entre as amostras em Y. O InfoNCE ainda assume que os rótulos são amostrados uniformemente, ou seja, $p_{\text{data}}(c) = \frac{1}{n}$. Na sequência, aplica-se o teorema de Bayes:

$$p_{\text{data}}(c|Y, y^*) = \frac{p_{\text{data}}(Y|c, y^*) p_{\text{data}}(c)}{\sum_{c'=1}^{n} p_{\text{data}}(Y|c', y^*) p_{\text{data}}(c')},$$

$$= \frac{p_{\text{pos}}(y_c|y^*) p_{\text{neg}}(y_c) \prod_{i=1}^{n} p_{\text{neg}}(y_i)}{\sum_{c'=1}^{n} p_{\text{pos}}(y_{c'}|y^*) p_{\text{neg}}(y_{c'}) \prod_{i=1}^{n} p_{\text{neg}}(y_i)},$$

$$= \frac{p_{\text{pos}}(y_c|y^*) p_{\text{neg}}(y_c)}{\sum_{c'=1}^{n} p_{\text{pos}}(y_{c'}|y^*) p_{\text{neg}}(y_{c'})}.$$

Um classificador ideal com entropia cruzada zero coincidiria com essa distribuição. A probabilidade ótima de uma classe é a razão de densidade $\frac{p_{\text{pos}}(y_c|y^*)}{p_{\text{neg}}(y_c)}$, normalizada em todas as classes. Isso descreve a probabilidade de y_c ser uma amostra positiva para y^* versus ser uma amostra negativa. Isso motiva a escolha do classificador do InfoNCE, que é definido de forma semelhante:

$$p_{\psi}(c|Y, y^*) = \frac{s_{\psi}(y^*, y_c)}{\sum_{c'=1}^{n} s_{\psi}(y^*, y_{c'})},$$

em que $s_{\psi}(y^*, y)$ é um preditor que calcula uma pontuação positivo real. Minimizar a entropia cruzada aproxima a distribuição do modelo $p_{\psi}(c|Y, y^*)$ à distribuição de dados $p_{\text{data}}(c|Y, y^*)$, o que garante que s_{ψ} se aproxime da razão de densidade dos dados, ou seja, $s_{\psi}(y^*, y) \approx \frac{p_{\text{pos}}(y|y^*)}{p_{\text{neg}}(y)}$, mas só precisa ser proporcional à razão de densidade.

A razão de densidade é alta para amostras positivas e próxima de zero para amostras negativas, o que significa que $s_{\psi}(y^*, y)$ aprende alguma medida de similaridade entre as representações. Como ψ e θ ou seja, preditor e codificador, são otimizados em conjunto, o codificador é incentivado a aprender incorporações semelhantes para uma âncora e seu positivo, e a aprender incorporações diferentes para uma âncora e suas amostras negativas. Em outras palavras, o codificador é incentivado a extrair informações que são "únicas"para a âncora e a amostra positiva. Esse objetivo maximiza a informação mútua entre y^* e y^+, que é um limite inferior para a informação mútua entre x^* e x^+.

A perda geral do InfoNCE para (y^*, Y, c) é definida como:

$$\text{InfoNCE}_{s_{\psi}}(y^*, Y, c) = -\log \left(\frac{s_{\psi}(y^*, y^+)}{\sum_{c'=1}^{n} s_{\psi}(y^*, y_{c'})} \right).$$

A perda do InfoNCE calcula a entropia cruzada softmax comumente usada. Em vez de especificar o rótulo de classe, denota-se o positivo por y^+ e o conjunto de negativos

por Y. Assim, a definição final da perda do InfoNCE para uma função de pontuação $s_\psi(y^*, y)$ é:

$$\text{InfoNCE}_{s_\psi}(y^*, y^+, Y) = -\log\left(\frac{\exp(s_\psi(y^*, y^+))}{\exp(s_\psi(y^*, y^+)) + \sum_{y \in Y} \exp(s_\psi(y^*, y))}\right).$$

O aprendizado auto-supervisionado contrastivo é uma técnica bastante usada para aprender representações de alta qualidade, focando a maximização da similaridade entre exemplos positivos e a minimização da similaridade entre exemplos negativos. A ideia central é que exemplos semelhantes devem ser representados próximos no espaço de representação, enquanto exemplos diferentes devem ser representados distantes. Essa abordagem tem sido aplicada com sucesso em diversas tarefas, incluindo classificação de imagens, detecção de objetos e reconhecimento de fala.

2.4.2. Contraste por instância de contexto

O **contraste por instância de contexto**, também conhecido como contraste global-local, é uma abordagem que visa modelar a relação entre a característica local de uma amostra e sua representação de contexto global. Quando o modelo aprende a representação para uma característica local, essa característica é associada à representação do conteúdo global, como nós para seus vizinhos. Existem dois tipos principais de Contraste de Contexto-Instância: *Previsão de Posição Relativa* e *Maximização de Informação Mútua*.

Previsão de posição relativa

A Previsão de Posição Relativa se concentra em aprender posições relativas entre componentes locais, em que o contexto global atua como um requisito implícito para prever essas relações. Muitos dados contêm ricas relações espaciais ou sequenciais entre suas partes. Vários modelos consideram o reconhecimento de posições relativas entre partes como a tarefa de pretexto [Jing e Tian, 2020]. Pode ser para prever as posições relativas de dois blocos menores (*patches*) a partir de uma amostra [Doersch et al., 2015], ou para recuperar as posições de segmentos embaralhados de uma imagem, resolver quebra-cabeças [Kim et al., 2018, Noroozi e Favaro, 2016, Wei et al., 2019], ou para inferir o grau de rotação de uma imagem [Gidaris et al., 2018]. A previsão de posição relativa também pode servir como ferramentas para criar amostras positivas difíceis. Por exemplo, a técnica do quebra-cabeça é aplicada no PIRL [Misra e Maaten, 2020] para aumentar a amostra positiva, mas o PIRL não considera resolver o quebra-cabeça e recuperar a relação espacial como seu objetivo.

Nos modelos de linguagem pré-treinados, ideias semelhantes, como a Previsão da Próxima Sentença (*Next Sentence Prediction* - NSP), também são adotadas. A perda de NSP foi inicialmente introduzida pelo BERT [Devlin et al., 2018], em que, para uma frase, o modelo deve distinguir a seguinte de uma amostrada aleatoriamente. No entanto, alguns trabalhos posteriores provam empiricamente que NSP ajuda pouco, ou até prejudica o desempenho. Portanto, no RoBERTa [Liu et al., 2019b], a perda de NSP é removida.

Para substituir a NSP, o ALBERT [Lan et al., 2019] propõe a tarefa de Previsão da Ordem das Sentenças (*Sentence Order Prediction* - SOP). No ALBERT, a tarefa de Previsão da Ordem das Sentenças (SOP) é uma estratégia de treinamento que visa melhorar a capacidade do modelo de capturar a coerência inter-sentença em um texto. Nessa tarefa, o modelo é apresentado com pares de segmentos de texto consecutivos de um documento e é treinado para prever a ordem correta desses segmentos.

Para realizar o treinamento, são utilizados exemplos positivos e negativos. Os exemplos positivos consistem em dois segmentos consecutivos do mesmo documento, mantendo a ordem original. Por outro lado, os exemplos negativos são compostos pelos mesmos dois segmentos, mas com a ordem trocada. Essa abordagem desafia o modelo a entender e capturar as relações de coerência entre as sentenças, em vez de simplesmente prever tópicos ou palavras isoladas.

Ao focara coerência inter-sentença, o ALBERT busca melhorar a capacidade do modelo de compreender o contexto global de um texto e a relação entre diferentes partes do mesmo. Isso é fundamental para tarefas de processamento de linguagem natural que exigem uma compreensão mais profunda do texto, como tradução automática, resumo de texto, análise de sentimentos e questionamento e resposta. A tarefa SOP no ALBERT atua como um mecanismo de treinamento eficaz para melhorar a representação de texto e a capacidade de modelagem de coerência textual.

Maximização de informação mútua

Este tipo de método deriva da informação mútua (*Mutual Information* - MI). A informação mútua visa modelar a associação entre duas variáveis. Geralmente, esses modelos otimizam $\max_{g_1 \in G_1, g_2 \in G_2} I(g_1(x_1), g_2(x_2))$, em que g_i é o codificador de representação, G_i é uma classe de codificadores com algumas restrições, e $I(\cdot,\cdot)$ é um estimador baseado em amostras para a informação mútua precisa. Na prática, MI é notória por sua computação complexa. Uma prática comum é maximizar alternativamente o limite inferior de I com um objetivo NCE.

Deep InfoMax (DIM) [Hassani e Khasahmadi, 2020] foi o primeiro a modelar explicitamente a informação mútua por meio de uma tarefa de aprendizado contrastivo, maximizando a MI entre um *patch* local e seu contexto global. Em classificação de imagens, por exemplo, pode-se codificar uma imagem de gato x em $f(x) \in \mathbb{R}^{M \times M \times d}$ e extrair um vetor de característica local $v \in \mathbb{R}^d$. Para conduzir o contraste entre instância e contexto, precisa-se de uma função resumo $g : \mathbb{R}^{M \times M \times d} \to \mathbb{R}^d$ para gerar o vetor de contexto $s = g(f(x)) \in \mathbb{R}^d$ e outra imagem de gato x^- e seu vetor de contexto $s^- = g(f(x^-))$.

Deep InfoMax proporciona um novo paradigma e impulsiona o desenvolvimento do aprendizado auto-supervisionado. Um modelo derivado do Deep InfoMax é o Contrastive Predictive Coding (CPC) [Oord et al., 2018] para reconhecimento de fala, que maximiza a associação entre um segmento de áudio e seu contexto. CPC também foi aplicado na classificação de imagens. DeepInfoMax de Escala Aumentada *Augmented Multiscale DIM - AMDIM* [Bachman et al., 2019] aprimora a associação positiva entre uma característica local e seu contexto, amostrando aleatoriamente duas visões diferentes de uma imagem para gerar o vetor de característica local e o vetor de contexto, respectivamente. CMC estende essa ideia para várias visões de uma imagem, amostrando outra

imagem irrelevante como negativa.

O Deep Graph Structural Infomax (DGSI) [Zhao et al., 2023] é uma extensão do Deep Graph InfoMax (DGI) que incorpora informações estruturais e topológicas para melhorar a representação dos nós de maneira auto-supervisionada. O DGSI consegue capturar informações semânticas mais detalhadas e informações estruturais benéficas utilizando várias abordagens. Uma dessas abordagens é a maximização da informação mútua estrutural. O DGSI aplica o princípio do *Information Bottleneck* para estabelecer restrições de informação mútua estrutural, equilibrando a suficiência e a minimalidade da representação ao preservar a estrutura topológica. Dessa forma, o modelo maximiza a informação mútua estrutural em relação às arestas e aos vizinhos locais, capturando detalhes finos da estrutura do grafo.

Além disso, o DGSI integra informações locais e globais ao maximizar a informação mútua entre a representação do nó e o resumo global do grafo, considerando também a conexão detalhada entre o nó e sua região receptiva local. Combinando essas restrições de informação mútua estrutural e representacional em uma única estrutura, o DGSI é capaz de capturar tanto a informação semântica dos nós quanto a informação estrutural do grafo, resultando em representações de nós mais ricas e abrangentes.

2.4.3. Contraste de instância para instância

O **aprendizado contrastivo de instância para instância** é uma abordagem que se concentra na modelagem da relação entre pares de instâncias, contrastando-as para aprender representações que capturem a estrutura subjacente dos dados. Ao contrário do contraste contexto-instância, que se concentra na relação entre uma amostra e seu contexto, essa técnica visa aprender representações discriminativas por meio da comparação direta de pares de instâncias. Essa abordagem motiva a rede a aprender representações discriminativas maximizando a similaridade entre pares de instâncias positivas e minimizando a similaridade entre instâncias negativas. Em outras palavras, o modelo é incentivado a aprender a distinguir entre diferentes instâncias, o que é fundamental para tarefas de classificação.

O aprendizado contrastivo de instância para instância é uma técnica poderosa para aprender representações discriminativas a partir de dados não rotulados. Ela tem demonstrado bons resultados em várias tarefas de aprendizado de máquina, especialmente em problemas de classificação linear, em que a capacidade de distinguir entre classes é essencial. Além disso, é uma técnica simples e eficaz que pode ser facilmente aplicada a uma ampla gama de problemas de aprendizado de máquina, proporcionando uma abordagem flexível e robusta para a aprendizagem de representações de dados.

2.4.4. Pré-treinamento contrastivo

Enquanto o aprendizado autos-supervisionado baseado em aprendizado contrastivo continua a ultrapassar limites em vários *benchmarks*, os rótulos ainda são importantes porque há uma lacuna entre os objetivos de treinamento do aprendizado auto-supervisionado e do aprendizado supervisionado. Em outras palavras, não importa o quanto os modelos de aprendizado auto-supervisionado melhorem, eles ainda são apenas extratores de características poderosos e, para transferir para a tarefa subsequente,

ainda há a necessidade de rótulos de alguma forma. Como resultado, para preencher a lacuna entre o pré-treinamento auto-supervisionado e as tarefas subsequentes, o aprendizado semi-supervisionado é um tipo de recurso empregado.

Aprendizado semi-supervisionado é uma abordagem de aprendizado de máquina que combina uma pequena quantidade de dados rotulados com muitos dados não rotulados durante o treinamento. Vários métodos derivam de diferentes suposições feitas sobre a distribuição dos dados, sendo o auto-treinamento o mais antigo. No auto-treinamento, um modelo é treinado com a pequena quantidade de dados rotulados e, em seguida, gera rótulos nos dados não rotulados. Apenas aqueles dados com rótulos altamente confiáveis são combinados com os dados rotulados originais para treinar um novo modelo. O processo ocorre de maneira iterativa até encontrar o melhor modelo.

O pré-treinamento contrastivo visa aprender representações úteis dos dados, enfatizando a similaridade entre instâncias positivas e reduzindo a similaridade entre instâncias negativas. Essa abordagem busca criar um espaço de representação em que instâncias semelhantes estejam próximas umas das outras e instâncias diferentes estejam distantes. Esse método é formulado através de uma função de perda, na qual as instâncias de entrada são representadas por x_i, suas representações latentes por $f(x_i)$, e $sim(x_i, x_j)$ é uma medida de similaridade entre elas. A função de perda é definida considerando o número total de instâncias N e uma temperatura τ que controla a suavização da distribuição de similaridade.

O método inclui a etapa de ajuste do modelo usando dados rotulados no processo de auto-treinamento semi-supervisionado. Esse processo envolve gerar pseudo-rótulos para dados não rotulados, usando o modelo treinado, e repetir o treinamento várias vezes até que o desempenho do modelo se estabilize. O pré-treinamento contrastivo tem sido aplicado com sucesso em tarefas de visão computacional, como classificação de imagens, detecção de objetos e segmentação semântica, com exemplos notáveis como SimCLR [Chen et al., 2020b], MoCo [He et al., 2020a] e BYOL [Grill et al., 2020].

À luz do sucesso do auto-treinamento semi-supervisionado, é natural repensar sua relação com os métodos auto-supervisionados, especialmente com os métodos de pré-treinamento contrastivo bem-sucedidos. Para tarefas de visão computacional, soluções como as apresentadas por [Zoph et al., 2020] estudam o pré-treinamento do MoCo e um método de auto-treinamento em que um professor é primeiro treinado em um conjunto de dados subsequente e depois gera pseudo-rótulos em dados não rotulados e finalmente um modelo aluno aprende conjuntamente sobre rótulos reais no conjunto de dados subsequente e pseudo-rótulos em dados não rotulados. O estudo aponta que o desempenho do pré-treinamento é prejudicado enquanto o auto-treinamento ainda se beneficia de um forte aumento de dados. Além disso, a adição de mais dados rotulados diminui o valor do pré-treinamento, enquanto o auto-treinamento semi-supervisionado sempre apresenta melhorias. Também foi identificado que as melhorias provenientes do pré-treinamento e do auto-treinamento são ortogonais entre si, ou seja, contribuem para o desempenho a partir de diferentes perspectivas. O modelo que combina pré-treinamento e auto-treinamento é o que apresenta o melhor desempenho.

O SimCLR de [Chen et al., 2020b] mostra que com apenas 10% dos rótulos originais do ImageNet [Deng et al., 2009], o ResNet-50 [Wen et al., 2020] pode superar o

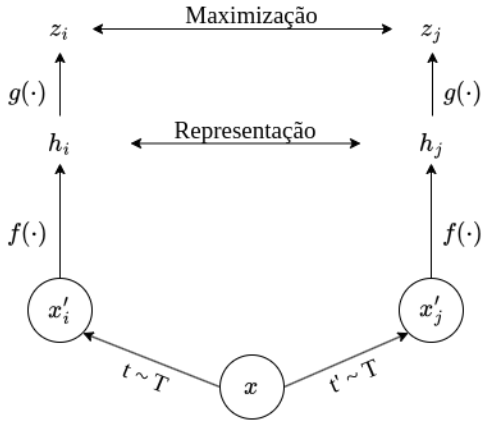

Figura 2.6. O arcabouço SimCLR. Duas operações de aumento de dados separadas são amostradas da mesma família de aumentações ($t \sim T$ e $t_0 \sim T$) e aplicadas a cada exemplo de dados para obter duas visualizações correlacionadas. Uma rede codificadora base $f(\cdot)$ e uma cabeça de projeção $g(\cdot)$ são treinadas para maximizar o acordo usando uma perda contrastiva. Após o treinamento ser concluído, descarta-se a cabeça de projeção $g(\cdot)$ e usa-se a codificadora $f(\cdot)$ e a representação h para tarefas posteriores. Adaptado de [Chen et al., 2020b].

supervisionado com pré-treinamento e auto-treinamento conjuntos. Eles propõem um arcabouço de 3 etapas:

1. Realizar pré-treinamento auto-supervisionado como o SimCLR v1, com algumas pequenas modificações na arquitetura e um ResNet mais profundo;
2. Ajustar as últimas camadas com apenas 1% ou 10% dos rótulos originais do ImageNet;
3. Usar a rede ajustada como professor para gerar rótulos em dados não rotulados para treinar um ResNet-50 aluno menor.

O sucesso na combinação de pré-treinamento auto-supervisionado contrastivo e auto-treinamento semi-supervisionado é uma tendência para o futuro paradigma de aprendizado profundo eficiente em dados. Mais trabalhos são esperados para investigar seus mecanismos latentes.

SimCLR - ***Simple Framework for Contrastive Learning of Visual Representations***

O arcabouço proposto por [Chen et al., 2020b] e apresentado na Figura 2.6 é similar a métodos anteriores como VICReg ou Barlow Twins. Dado um lote de imagens X, duas visualizações $X^{(1)} = t(X)$ e $X^{(2)} = t(X)$ são criadas usando transformações aleatórias $t \sim T$. Um codificador Siamese f_θ calcula representações $Y^{(1)} = f_\theta(X^{(1)})$ e $Y^{(2)} = f_\theta(X^{(2)})$, que são então alimentadas em um projetor Siamese g_ϕ para obter projeções $Z^{(1)} = [z_1^{(1)}, \ldots, z_n^{(1)}] = g_\phi(Y^{(1)})$ e $Z^{(2)} = [z_1^{(2)}, \ldots, z_n^{(2)}] = g_\phi(Y^{(2)})$. A Figura 2.6 mostra uma visão geral do processo.

O SimCLR usa uma perda contrastiva para maximizar a similaridade entre as duas projeções da mesma imagem enquanto minimiza a similaridade com as projeções de outras imagens. Especificamente, para uma imagem x_i, são aplicadas duas perdas InfoNCE.

A primeira usa o âncora $z_i^{(1)}$, o positivo $z_i^{(2)}$ e os negativos $\bar{Z}_i = [z_1^{(1)}, z_1^{(2)}, \ldots, z_n^{(1)}, z_n^{(2)}] \setminus \{z_i^{(1)}, z_i^{(2)}\}$, que são todas projeções de outras imagens no lote. A segunda perda InfoNCE troca os papéis de âncora e positivo, mas usa o mesmo conjunto de negativos. Portanto, a função de perda é definida como

$$L_{\text{SimCLR}}^{\theta,\phi} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(\text{InfoNCE}_{s_\tau}(z_i^{(1)}, z_i^{(2)}, \bar{Z}_i) + \text{InfoNCE}_{s_\tau}(z_i^{(2)}, z_i^{(1)}, \bar{Z}_i)\right),$$

em que as similaridades são calculadas como $s_\tau(z, z') = \text{scos}(z, z')/\tau$, ou seja, a similaridade cosseno dividida por um hiperparâmetro de temperatura $\tau > 0$.

As transformações consistem em um recorte aleatório seguido por um redimensionamento de volta ao tamanho original, uma distorção de cor aleatória e um desfoque gaussiano aleatório. Um ResNet é usado como codificador f_θ e o projetor g_ϕ é implementado como uma MLP com uma camada oculta. Para treinar o SimCLR, são utilizados tamanhos de lote grandes em combinação com o otimizador LARS. Os autores observam que seu método não precisa de bancos de memória, como é o caso de outros métodos contrastivos, e é portanto mais fácil de implementar.

MOCO - *Momentum Contrast*

O Momentum Contrast [He et al., 2020a] é uma abordagem de aprendizado contrastivo que utiliza um codificador de *momentum* com uma fila de codificação. Em essência, ele permite a otimização de um objetivo contrastivo com custos computacionais significativamente reduzidos, tanto em termos de tempo quanto de memória da GPU [Chen et al., 2020a]. Conforme mostrado na Figura 2.7, MoCo define uma rede de aluno consistindo de um codificador f_θ e um projetor g_ϕ com parâmetros θ e ϕ, e uma rede de professor consistindo de um codificador $f_{\bar{\theta}}$ e um projetor $g_{\bar{\phi}}$ com parâmetros $\bar{\theta}$ e $\bar{\phi}$. Dada uma imagem x_i, duas visualizações $x_i^* = t(x_i)$ e $x_i^+ = t(x_i)$ são criadas usando transformações aleatórias $t \sim T$. O aluno computa a representação $y_i^* = f_\theta(x_i^*)$ e a projeção $z_i^* = g_\phi(y_i^*)$, enquanto o professor computa a representação $y_i^+ = f_{\bar{\theta}}(x_i^+)$ e a projeção $z_i^+ = g_\phi(y_i^+)$. O MoCo minimiza a perda do InfoNCE para aprender projeções que sejam semelhantes para duas visualizações da mesma imagem e diferentes das projeções de visualizações de outras imagens. A perda do MoCo é então definida como:

$$L_{MoCo}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n}\text{InfoNCE}_\tau(z_i^*, z_i^+, Z_i^-),$$

em que as similaridades são calculadas usando o produto escalar $s_\tau(z^*, z) = z^\top z^*/\tau$ dividido por um hiperparâmetro de temperatura $\tau > 0$. O professor é atualizado por uma média móvel exponencial do estudante, ou seja,

$$\bar{\theta} \leftarrow \alpha\bar{\theta} + (1-\alpha)\theta$$
$$\bar{\phi} \leftarrow \alpha\bar{\phi} + (1-\alpha)\phi$$

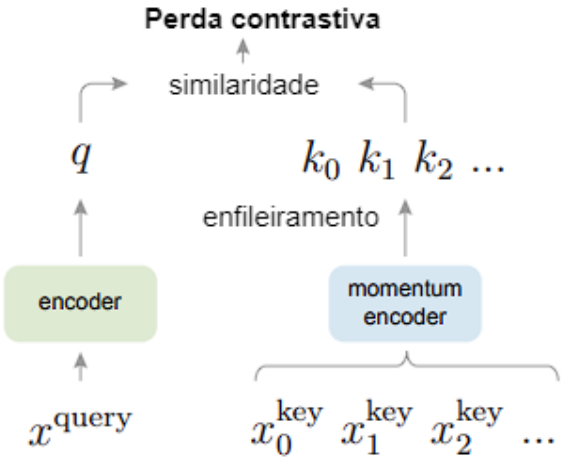

Figura 2.7. O MoCo treina um codificador de representações visuais combinando uma consulta codificada q a um dicionário de chaves codificadas usando uma perda contrastiva. O dicionário é dinamicamente definido pelas amostras de dados e construído como uma fila, permitindo um dicionário grande e consistente para aprender representações visuais. As chaves são codificadas por um codificador de progressão lenta, impulsionado por uma atualização de *momentum* com o codificador de consulta. Adaptado de [He et al., 2020a].

em que $\alpha \in [0,1]$ controla a taxa na qual os pesos da rede do professor são atualizados com os pesos da rede do estudante. O MoCo v2.0 [Chen et al., 2020a] introduz várias mudanças menores para melhorar ainda mais o desempenho *downstream* e superar o SimCLR. As mudanças mais notáveis incluem a substituição da camada de projeção linear do MoCo por um MLP, bem como a aplicação de um agendador de taxa de aprendizado cosseno e modificações adicionais. O novo cabeçalho MLP de 2 camadas foi adotado seguindo o SimCLR. Note-se que o MLP é usado apenas durante o treinamento não supervisionado e não é destinado a tarefas *downstream*. Em termos de aumentos de dados adicionais, o MoCo v2.0 também adota a operação de desfoque usada no SimCLR.

PIRL - *Pretext-Invariant Representation Learning*

Nas tarefas de pretexto, calculam-se representações de imagens transformadas para prever propriedades específicas, como ângulos de rotação ou permutações de *patches*. Embora isso incentive a covariância às transformações, o foco é representações semanticamente significativas e invariantes à transformação. Para isso, [Misra e Maaten, 2020] desenvolveram o PIRL, uma abordagem que refinou a formulação da perda da tarefa de pretexto e utiliza bancos de memória. A representação do modelo PIRL está expressa na Figura 2.8.

O objetivo do PIRL é treinar uma rede codificadora f_θ que mapeia imagens $x_i^{(1)} = x_i$ e imagens transformadas $x_i^{(2)} = t_\pi(x_i)$ para representações $y_i^{(1)}$ e $y_i^{(2)}$, respectivamente, que são invariantes às transformações utilizadas. Nesse caso, t_π denota uma transformação de quebra-cabeça, consistindo em uma permutação aleatória de *patches* de imagem, em que π é a permutação correspondente. A formulação da perda das tarefas de pretexto, enfatiza que o codificador aprende representações que contêm informações sobre a transformação, e não sobre a semântica. Sejam $z_i^{(1)} = g_\phi(f_\theta(x_i^{(1)}))$ e

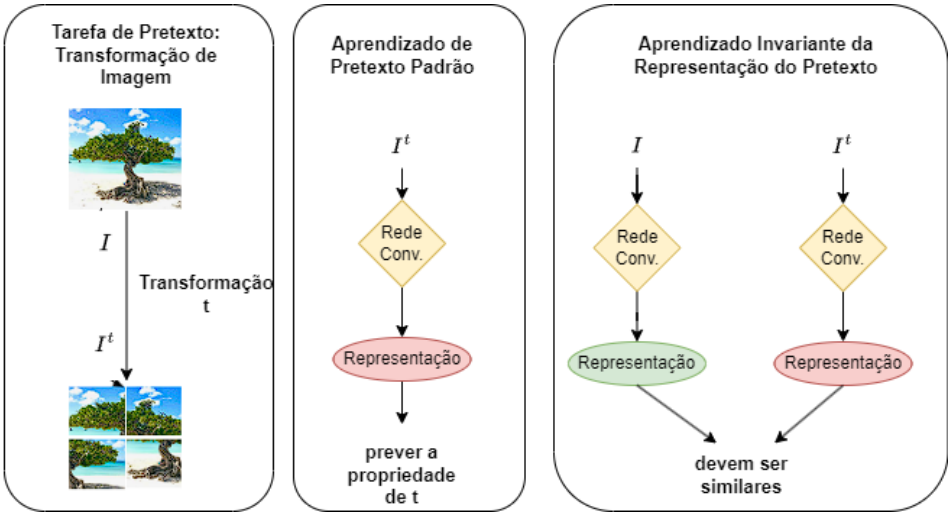

Figura 2.8. Muitas tarefas de pretexto no aprendizado auto-supervisionado envolvem transformar uma imagem, calcular sua representação transformada e prever propriedades da transformação. Enquanto essas representações geralmente variam com a transformação e podem ter pouca informação semântica, o PIRL aprende representações que são invariantes à transformação e mantêm informações semânticas. Adaptado de [Misra e Maaten, 2020].

$z_i^{(2)} = g_\psi(f_\theta(x_i^{(2)}))$ as projeções obtidas pelo codificador f_θ e dois projetores separados g_ϕ e g_ψ. A rede é treinada minimizando uma combinação convexa de dois estimadores contrastivos de ruído (NCE) [Gutmann e Hyvärinen, 2010]:

$$\mathscr{L}_{PIRL}(\theta,\phi,\psi) = \frac{1}{n}\sum_{i=1}^{n} \lambda \mathscr{L}_{NCE}(m_i, z_i^{(2)}, \bar{M}_i) + (1-\lambda)\mathscr{L}_{NCE}(m_i, z_i^{(1)}, \bar{M}_i)$$

em que m_i é uma projeção de um banco de memória correspondente à imagem original x_i, cada amostra positiva é atribuída a um conjunto aleatório de projeções negativas $\bar{M}_i$ de imagens diferentes de x_i obtidas do banco de memória, e $\lambda \in [0,1]$ é um hiperparâmetro. Em contraste com as tarefas de pretexto introduzidas anteriormente, a formulação da perda do PIRL não visa explicitamente prever propriedades particulares das transformações aplicadas, como rotação ou índices de *patches*. Em vez disso, é definida apenas em imagens e suas contrapartes transformadas correspondentes. O NCE aplica classificação binária a cada ponto de dados para distinguir amostras positivas e negativas. Nesse caso, a perda NCE é formulada como:

$$\mathscr{L}_{NCE}(m,z,\bar{M}) = -\log[h(m,z,\bar{M})] - \sum_{\bar{m}\in\bar{M}} \log[1-h(z,m,\bar{M})]$$

em que h modela a probabilidade de que (x_i, x_i') seja derivado de X como:

$$h(u,v,\bar{M}) = \frac{\exp(\text{scos}(u,v)/\tau)}{\exp(\text{scos}(u,v)/\tau) + \sum_{\bar{m}\in\bar{M}} \exp(\text{scos}(\bar{m},v)/\tau)}.$$

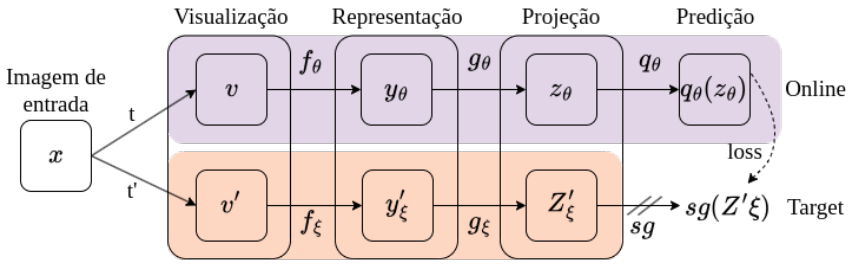

Figura 2.9. Arquitetura do Modelo BYOL. O BYOL minimiza uma perda de similaridade entre $q_\theta(z_\theta)$ e $sg(z_{\xi_0})$, em que θ são os pesos treinados, ξ é uma média móvel exponencial de θ e sg significa *stop-gradient*. Adaptado de [Grill et al., 2020].

BYOL - *Bootstrap Your Own Latent*

O BYOL [Grill et al., 2020] é inspirado na observação de que aprender representações ao prever representações fixas de uma rede alvo inicializada aleatoriamente evita o colapso da representação, apesar de apresentar desempenho inferior. Isso naturalmente implica uma arquitetura de professor-aluno, em que o professor fornece representações estáveis para o aluno aprender.

BYOL define duas redes diferentes: uma rede aluno e uma rede professor. A arquitetura é mostrada na Figura 2.9, a rede aluno e a rede professor consistem das seguintes partes:

- Rede aluno: codificador f_θ, projetor g_ϕ, preditor q_ψ
- Rede professor: codificador $f_{\bar{\theta}}$, projetor $g_{\bar{\phi}}$

O codificador f e o projetor g estão presentes tanto nas redes aluno quanto nas professor, enquanto o preditor q faz parte apenas da rede aluno.

Como os métodos de maximização de informação, os métodos professor-aluno aprendem representações aplicando diferentes transformações às imagens. Dada uma imagem x_i, o BYOL aplica transformações amostradas aleatoriamente $t \sim T$ para obter duas visualizações diferentes $x_{(1)i} = t(x_i)$ e $x_{(2)i} = t(x_i)$. A rede aluno calcula representações $y_{(j)i} = f_\theta(x_{(j)i})$, projeções $z_{(j)i} = g_\phi(y_{(j)i})$ e previsões $\hat{z}_{(j)i} = q_\psi(z_{(j)i})$ para ambas as visualizações $j \in \{1,2\}$. As visualizações também são alimentadas na rede professor para obter projeções alvo $\bar{z}_{(1)i} = g_{\bar{\phi}}(f_{\bar{\theta}}(x_{(1)i}))$ e $\bar{z}_{(2)i} = g_{\bar{\phi}}(f_{\bar{\theta}}(x_{(2)i}))$.

BYOL minimiza dois erros quadrados normalizados:

- entre a previsão da primeira visualização e a projeção alvo da segunda visualização
- entre a previsão da segunda visualização e a projeção alvo da primeira visualização

A função de perda final é:

$$L_{\text{BYOL}}^{\theta,\phi,\psi} = \frac{1}{n}\sum_{i=1}^{n}\left[\text{dnse}(\hat{z}_{(1)i}, \bar{z}_{(2)i}) + \text{dnse}(\hat{z}_{(2)i}, \bar{z}_{(1)i})\right].$$

A perda é mínima quando a similaridade de cosseno entre os vetores é 1. Assim, são aprendidas representações que são semelhantes para duas transformações diferentes.

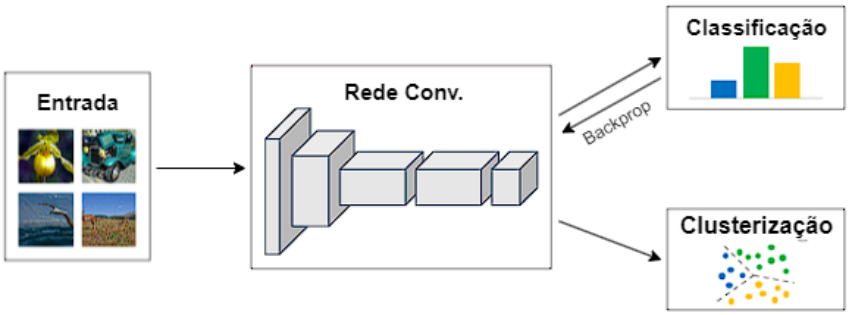

Figura 2.10. Modelo SimSiam em que características profundas são agrupadas iterativamente e as atribuições de cluster são usadas como pseudo-rótulos para aprender os parâmetros da rede convolucional. Adaptado de [Chen e He, 2021].

Em outras palavras, o conteúdo de informação nas representações aprendidas é maximizado.

Em cada etapa de treinamento, a perda é minimizada em relação a θ, ϕ e ψ. Ou seja, apenas os pesos do aluno são atualizados pelo gradiente da função de perda usando o otimizador LARS. Os pesos do professor são atualizados pela média móvel exponencial, ou seja,

$$\bar{\theta} \leftarrow \tau\bar{\theta} + (1-\tau)\theta, \quad \bar{\phi} \leftarrow \tau\bar{\phi} + (1-\tau)\phi,$$

em que $\tau \in [0,1]$ controla a taxa na qual os pesos da rede professor são atualizados com os pesos da rede aluno.

SimSiam - *Simple Siamese Representation Learning*

O SimSiam utiliza uma arquitetura e função de perda similares ao BYOL. No entanto, o professor e o aluno compartilham os mesmos parâmetros e, portanto, um codificador de *momentum* não é usado como nos métodos professor-aluno apresentados anteriormente.

Dado um lote de imagens X, para cada imagem x_i, duas visualizações $x_{(1)i} = t(x_i)$ e $x_{(2)i} = t(x_i)$ são criadas usando transformações aleatórias $t \sim T$ que são amostradas para cada imagem e cada visualização. Para cada uma dessas visualizações, um codificador Siamese f_θ calcula uma representação $y_{(j)i} = f(x_{(j)i})$ e um projetor Siamese g_ϕ calcula uma projeção $z_{(j)i} = g_\phi(y_{(j)i})$. Finalmente, a projeção é alimentada através de um preditor q_ψ para obter uma previsão $\hat{z}_{(j)i} = q_\psi(z_{(j)i})$.

O objetivo do preditor é prever a projeção da outra visualização. Portanto, a perda calcula a similaridade cosseno negativa entre a previsão da primeira visualização e a projeção da segunda visualização, e vice-versa, ou seja,

$$L_{\text{SimSiam}}^{\theta,\phi,\psi} = -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(\text{scos}(\hat{z}_{(1)i}, \text{sg}(z_{(2)i})) + \text{scos}(\hat{z}_{(2)i}, \text{sg}(z_{(1)i}))\right),$$

em que $\text{sg}(\cdot)$ é o operador de *stop-gradient* que impede que os gradientes sejam retropropagados por esse ramo do grafo computacional.

O codificador f_θ é implementado como uma ResNet. O projetor g_ϕ e o preditor q_ψ são MLPs. Os autores mostram empiricamente que um preditor é crucial para evitar o colapso. Os autores [Chen e He, 2021] argumentam que o gradiente da perda simetrizada com um preditor que é a identidade está na mesma direção que o gradiente da perda simetrizada entre as duas projeções, de modo que a operação de *stop-gradient* é cancelada, levando assim ao colapso da representação. Usar um preditor aleatório também não funciona e [Chen e He, 2021] argumentam que o preditor deve sempre aprender as representações mais recentes.

Outro ingrediente importante para o método é a normalização em lote, que é usada tanto para f_θ quanto para g_ϕ. Além disso, os autores experimentam com o objetivo de treinamento substituindo-o pela perda de entropia cruzada. Suas experiências mostram que isso também funciona, no entanto, o desempenho é pior. A principal vantagem do SimSiam é que o treinamento não requer lotes grandes, permitindo o uso de *Stochastic Gradient Descent*.

2.5. Aprendizado baseado em *clusters*

Outra estratégia, denominada *Cluster Discrimination*, parte do princípio de que amostras do mesmo agrupamento (*cluster*) devem ser representadas próximas umas das outras, enquanto amostras de agrupamentos diferentes devem ter representações distantes. Essa abordagem tem demonstrado alto desempenho em tarefas subsequentes, especialmente em problemas de classificação linear. No entanto, a eficácia dessas técnicas pode depender de fatores como a escolha apropriada de hiperparâmetros e a qualidade dos dados de treinamento. Portanto, ao implementar essas abordagens, é crucial considerar esses aspectos para garantir resultados satisfatórios em diversas aplicações de aprendizado de máquina.

Classificação de imagens pede que o modelo categorize imagens corretamente e a representação de imagens na mesma categoria deve ser semelhante. Portanto, a motivação é aproximar imagens similares no espaço de incorporação (*embedding*). No aprendizado supervisionado, esse processo de aproximação é realizado por meio da supervisão de rótulos; no entanto, no aprendizado auto-supervisionado, não há rótulos. Para resolver o problema dos rótulos, DeepCluster [Caron et al., 2018] propõe utilizar clusterização para gerar pseudo-rótulos e pede a um discriminador para prever os rótulos das imagens. O treinamento pode ser formulado em duas etapas. Na primeira etapa, o DeepCluster usa K-means para agrupar a representação codificada e produz pseudorrótulos para cada amostra. Então, na segunda etapa, o discriminador prevê se duas amostras são do mesmo cluster e realiza o backpropagation para o codificador. Essas duas etapas são realizadas iterativamente.

O Local Aggregation (LA) [Zhuang et al., 2019] avançou os limites do método baseado em clusterização. Ele aponta várias desvantagens do *DeepCluster* e faz as correspondentes otimizações. Primeiro, no *DeepCluster*, as amostras são atribuídas a clusters mutuamente exclusivos, mas o LA identifica vizinhos separadamente para cada exemplo. Segundo, o *DeepCluster* otimiza uma perda discriminativa de entropia cruzada, enquanto o LA emprega uma função objetivo que otimiza diretamente uma métrica de soft-clustering local. Essas duas mudanças aumentam substancialmente o desempenho

da representação do LA em tarefas subsequentes.

Um trabalho semelhante ao LA é o VQ-VAE [Razavi et al., 2019] para superar a deficiência tradicional do VAE de gerar imagens de alta fidelidade, o VQ-VAE propõe quantizar vetores. Para a matriz de características codificada a partir de uma imagem, o VQ-VAE substitui cada vetor unidimensional na matriz pelo mais próximo em um dicionário de incorporação. Esse processo é de certa forma semelhante ao que o LA está fazendo. Modelos alternativos ao VQ-VAE, como proposto por [Peng et al., 2021], usa um módulo de atenção estrutural dentro da rede de geração de textura, em que o módulo utiliza a informação estrutural para capturar correlações distantes. Desta forma, reutlizando o VQ-VAE para calcular duas perdas de características, que ajudam a melhorar a coerência da estrutura e o realismo da textura, respectivamente.

Apesar do sucesso anterior do aprendizado contrastivo baseado em discriminação de clusters, o paradigma de treinamento em duas etapas é demorado e de baixo desempenho comparado aos métodos posteriores baseados em discriminação de instâncias, incluindo CMC [Tian et al., 2020], MoCo [He et al., 2020b] e SimCLR [Chen et al., 2020b]. Esses métodos baseados em discriminação de instâncias eliminaram a etapa de agrupamento lento e introduziram estratégias eficientes de *data augmentation* para aumentar o desempenho. Em virtude desses problemas, os autores do SwAV [Caron et al., 2020] trazem ideias de realizar clusterização *online* e estratégias de *data augmentation* para a abordagem de discriminação de clusters. O SwAV propõe objetivos contrastivos de predição trocada para lidar com o aumento de dados *multiview*. A intuição é que, dados alguns protótipos (agrupados), diferentes visualizações das mesmas imagens devem ser atribuídas aos mesmos protótipos. O SwAV chama essa “atribuição"de “códigos". Para acelerar o cálculo dos códigos, os autores do SwAV desenvolvem uma estratégia de cálculo *online*. Com base no SwAV, um modelo auto-supervisionado [Goyal et al., 2021] com 1,3 bilhão de parâmetros foi treinado em 1 bilhão de imagens da web coletadas do Instagram.

2.5.1. *DeepCluster*

Um dos primeiros métodos a implementar a ideia de agrupamento para aprendizado de representação é o DeepCluster [Caron et al., 2018]. Ele alterna entre a criação de pseudo-rótulos via atribuições de clusters e o ajuste da representação para classificar imagens de acordo com seus rótulos inventados. A motivação por trás disso é aumentar o desempenho de arquiteturas convolucionais que já exibem um forte viés indutivo, já que essas tendem a se sair razoavelmente bem com pesos inicializados aleatoriamente. No geral, os autores propõem alternar repetidamente entre as seguintes duas etapas para melhorar ainda mais a rede codificadora:

1. Agrupar as representações $y_i = f_\theta(x_i)$ produzidas pelo estado atual do codificador f_θ em k clusters (por exemplo, usando agrupamento k-means);

2. Usar as atribuições de clusters da etapa 1 como pseudo-rótulos β_i para supervisão e atualizar os pesos, ou seja,

$$L_{\text{DeepCluster}}(\theta, \psi) = \frac{1}{n} \sum_{i=1}^{n} d_{\text{classification}}(q_\psi(y_i), \beta_i),$$

em que uma rede preditora q_ψ tenta prever as atribuições de cluster das representações $y_i = f_\theta(x_i)$.

Em seus experimentos, os autores utilizam uma rede AlexNet padrão [Krizhevsky et al., 2012] com K-means.

2.5.2. SwAV - *Swapping Assignments Between Multiple Views of the Same Image*

[Caron et al., 2020] propõem um algoritmo alternativo, chamado SwAV, que promove ao mesmo tempo consistência entre as atribuições de clusters em diferentes visualizações. Ao contrário do DeepCluster, o SwAV é uma abordagem de agrupamento *online*, ou seja, não alterna entre uma atribuição de cluster e uma etapa de treinamento. Uma rede codificadora f_θ é usada para calcular as representações de imagem $y^{(1)}$ e $y^{(2)}$ de duas visualizações da mesma imagem x. Essas representações são então mapeadas para um conjunto de protótipos parametrizados $C_\psi = [c_1, \ldots, c_k]$, resultando em códigos correspondentes $q^{(1)}$ e $q^{(2)}$. Em seguida, é abordado um problema de previsão trocada, em que os códigos derivados de uma visualização são previstos usando a codificação da segunda visualização. Para alcançar isso, minimiza-se L_{SwAV}, em que $\ell(q,y) = dce(q, \text{softmax}_\tau(C^\top y))$ quantifica a correspondência entre a representação y e o código q para uma temperatura $\tau > 0$. Embora o SwAV se beneficie do aprendizado contrastivo, não requer o uso de um grande banco de memória ou uma rede de *momentum*.

Além deste método, os autores também propõem a técnica de aumento chamada multi-crop, que também foi usada para DINO. Em vez de usar duas visualizações com resolução completa, é usada uma mistura de visualizações com diferentes resoluções. Nesta abordagem, múltiplas transformações são comparadas usando transformações consideravelmente menores, o que leva a uma melhoria adicional de métodos anteriores como SimCLR e DeepCluster.

2.6. Casos de uso para detecção de intrusões e provisão de QoS

Esta seção explora o uso do aprendizado auto-supervisionado em aplicações de redes de computadores dinâmicas, focando a detecção de intrusão em dispositivos IoT e na provisão de Qualidade de Serviço (QoS). Com o aumento de dispositivos conectados, as redes enfrentam desafios contínuos em segurança e eficiência. A detecção de intrusão é essencial para identificar e mitigar ameaças cibernéticas em tempo real, enquanto a provisão de QoS assegura que serviços críticos mantenham desempenho e confiabilidade em ambientes dinâmicos. O aprendizado auto-supervisionado utiliza grandes volumes de dados não rotulados para aprender representações úteis e tomar decisões inteligentes, sendo uma abordagem promissora para enfrentar esses desafios. A seção discute métodos inovadores e resultados que demonstram o uso dessas técnicas.

2.6.1. Detecção de intrusões em Internet das Coisas

A detecção de intrusões em Internet das Coisas (IoT) beneficia-se significativamente do uso de aprendizado auto-supervisionado, aproveitando-se da habilidade dessa abordagem de manipular grandes volumes de dados não rotulados [Barbosa et al., 2024]. Neste contexto, aprendizagem auto-supervisionado permite a captura de características essenciais e padrões comportamentais sem a necessidade de supervisão explícita, o que é ideal em cenários de IoT, onde etiquetar dados pode ser impraticável. Com modelos pré-treinados neste vasto conjunto de informações, o ajuste fino requer apenas um volume reduzido de dados rotulados, tornando o processo não apenas econômico, mas também mais ágil na resposta a possíveis ameaças. Treinar os modelos nesta configuração pode não apenas melhorar a eficácia do modelo na detecção de intrusões em ambientes IoT, mas também aprimorar a segurança em outros tipos de redes, evidenciando a versatilidade e abrangência dessa abordagem.

FeCo - ***Federated Contrastive Learning Framework***

O trabalho de [Wang et al., 2022a] apresenta uma solução para melhorar a capacidade de detecção de intrusão em redes IoT por meio de aprendizado contrastivo federado. A solução proposta é o FeCo, um arcabouço de aprendizado contrastivo federado que coordena dispositivos IoT na rede para aprender modelos de detecção de intrusão de forma conjunta. O FeCo emprega técnicas de aprendizado profundo e contrastivo para extrair e comparar representações de dados de tráfego de rede, identificando padrões anômalos que possam indicar atividades suspeitas. A implementação do FeCo envolve a colaboração entre os dispositivos de IoT, que compartilham localmente informações sobre o tráfego de rede com um agregador de modelo central.

Conforme mostrado na Figura 2.11, o diagrama de fluxo do FeCo inicia com a extração de características dos dados de tráfego de rede dos dispositivos de IoT, seguido pelo aprendizado contrastivo para comparar e distinguir entre padrões benignos e maliciosos. Posteriormente, o modelo treinado é utilizado para a detecção em tempo real de atividades suspeitas. A implementação do aprendizado federado permite a melhoria contínua do modelo, sem comprometer a privacidade dos dados dos dispositivos individuais, resultando em um sistema eficaz de detecção de intrusões para redes de IoT.

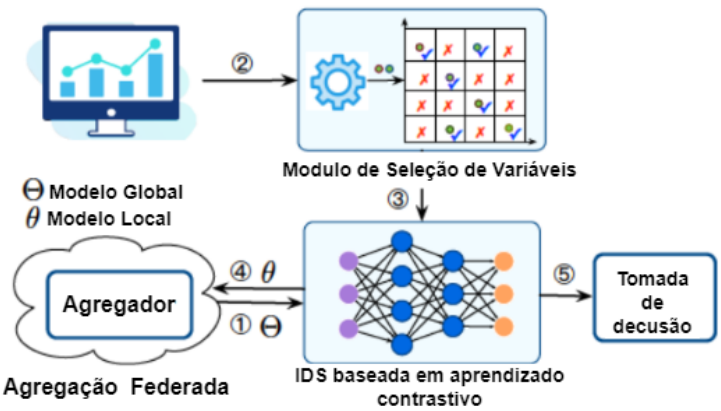

Figura 2.11. Diagrama de fluxo da proposta FeCo. A proposta implementa o aprendizado contrastivo federado. Inspirado em [Wang et al., 2022a].

O IDS baseado em aprendizado contrastivo é o bloco de construção do FeCo. O

objetivo ao implantar o aprendizado contrastivo é treinar um modelo que produza representações semelhantes para todas as instâncias normais de tráfego e torne as representações de intrusão distantes das representações normais. Especificamente, o aprendizado contrastivo treina um modelo de rede neural artificial (*Artificial Neural Network* - ANN) que recebe $x_i \in \mathbb{R}^d$ como entrada e gera uma nova representação $z_i \in \mathbb{R}^o$. O modelo de ANN pode ser representado por uma função $f_\theta : \mathbb{R}^d \to \mathbb{R}^o$, em que θ denota os parâmetros do modelo. O modelo de ANN do FeCo consiste em quatro camadas: a camada de entrada, duas camadas ocultas e, finalmente, a camada de saída.

Como o objetivo do aprendizado contrastivo é maximizar a similaridade entre v_i e v_j e minimizar a similaridade entre v_i e u_m para $m \in [M]$ (define-se $[M] := \{1, 2, ..., M\}$). Para atingir esse objetivo, usa-se uma função de perda L_{ij}, dada por:

$$L_{ij} = -\log \frac{\exp(v_i^T v_j/\tau)}{\exp(v_i^T v_j/\tau) + \sum_{m=1}^{M} \exp(v_i^T u_m/\tau)}, \quad ,$$

em que $\tau \in [0,1]$ representa um parâmetro de temperatura, $v_i = f_\theta(x_i)$ representa a saída de uma entrada benigna x_i e $u_i = f_\theta(x_i)$ a saída de uma entrada de intrusão x_i. Após o treinamento, para medir a similaridade $S(x_{\text{test}_j})$ entre um fluxo de tráfego iminente x_{test_j} e o *template* normalizado utiliza-se o estimador de similaridade cosseno:

$$S(x_{\text{test}_j}) = \frac{\bar{z}^T f_\theta(x_{\text{test}_j})}{\|\bar{z}\| \times \|f_\theta(x_{\text{test}_j})\|}.$$

O escore de similaridade $S(x_{\text{test}_j})$ varia de 0 a 1. É necessário haver um limiar $0 \leq \rho \leq 1$ para determinar se x_{test_j} é uma anomalia ou não. O FeCo foi construído incorporando o IDS baseado em aprendizado por contraste no arcabouço de aprendizado federado. Nesse caso, cada cliente participa no processo de Aprendizado Federado (FL) fornecendo sua atualização de parâmetro de modelo. Para isso, utilizaram o algoritmo FedAVG [McMahan et al., 2017] para agregar as atualizações de múltiplos clientes. No passo de tempo t, o agregador de modelo A computa o modelo global Θ_t por:

$$\Theta_t = \Theta_{t-1} + \sum_i c_i \cdot (\theta_i - \Theta_{t-1}),$$

em que θ_i são os parâmetros de modelo locais no cliente i e c_i é um coeficiente de peso. No caso da solução proposta, c_i é baseado no tamanho do conjunto de dados de treinamento local no cliente i. Especificamente, c_i é definida como a razão entre o tamanho do conjunto de dados de treinamento local no cliente i e o número total de amostras de treinamento em todos os clientes selecionados.

O desempenho do FeCo foi avaliado em comparação com outros modelos de detecção de intrusões em redes de IoT, utilizando o conjunto de dados NSL-KDD. Entre os modelos de *benchmark* utilizados estão o *Support Vector Machine* (SVM), o *Random Forest* (RF), o *Multi-Layer Perceptron* (MLP), o *Deep Neural Network* (DNN), o *Autoencoder* (AE), o *Variational Autoencoder* (VAE) e o *Generative Adversarial Network* (GAN). Os resultados mostraram que o FeCo superou outros modelos em termos de detecção de

intrusões, com uma taxa de detecção de 99,2% e uma taxa de falsos positivos de 0,8%. Além disso, o FeCo apresentou uma redução significativa na sobrecarga de comunicação em comparação com outros modelos, com uma redução de 99,9% na quantidade de dados transmitidos durante o treinamento. O FeCo também demonstrou sua escalabilidade, sendo capaz de lidar com um grande número de dispositivos de IoT em uma rede. Esses resultados indicam que o FeCo é uma solução eficaz e viável para aprimorar a segurança em redes de IoT.

Aprendizado Contrastivo sobre Características de Fourier Aleatórias

[Lopez-Martin et al., 2023] utilizam aprendizado contrastivo e características de Fourier aleatórias (RFFs). Conforme mostrado na Figura 2.12, a técnica mapeia as características da rede e os rótulos para um espaço comum, em que é possível medir a similaridade para realizar a classificação. O modelo é especialmente otimizado para identificar ataques desconhecidos, utilizando técnicas de regularização L2 e contrastiva para evitar o sobreajuste. Testes com os conjuntos de dados públicos demonstram que o modelo proposto supera as alternativas existentes na detecção de novos ataques, podendo ser executado em dispositivos de baixos recursos. A diversidade intra-classe e similaridade inter-classe em tráfego de rede são abordados em [Yue et al., 2022]. O método utiliza mascaramento aleatório de sequências de pacotes de rede para criar tarefas contrastivas, calculando a perda contrastiva para medir distâncias intra-classe e inter-classe. Experimentos com conjuntos de dados reais e de *benchmark* mostram melhorias significativas na precisão e na taxa de detecção de intrusões em ambientes complexos de rede. De forma semelhante, [Li et al., 2024a] utiliza o aprendizado contrastivo para melhorar a distinção entre tráfego benigno e malicioso no espaço de representação, resultando em maior precisão na detecção de ataques desconhecidos.

Detecção de Anomalias em Redes com Aprendizado Auto-Supervisionado

A proposta AOC-IDS visa a detecção de intrusões em tempo real em ambientes em que os comportamentos dos sistemas e as estratégias de ataque evoluem constantemente[Zhang et al., 2024]. O AOC-IDS integra um módulo de detecção de anomalias (ADM) e um arcabouço em tempo real que permite adaptação contínua. O ADM utiliza um *autoencoder* (AE) com uma função de perda contrastiva personalizada, chamada *Cluster Repelling Contrastive (CRC) loss*, que melhora a capacidade de representação dos dados. A estrutura online do AOC-IDS gera pseudo-rótulos automaticamente para atualizar periodicamente o ADM, eliminando a necessidade de intervenção humana para rotulagem e facilitando a adaptação autônoma do sistema a novos dados sem rótulos. De forma semelhante, o ContraMTD é um método não supervisionado para detecção de tráfego malicioso, também baseado em aprendizado contrastivo[Han et al., 2024]. O ContraMTD extrai características de comportamento local e interação global do tráfego normal, utilizando aprendizado contrastivo para aprender a relação entre elas e detectar anomalias. O ContraMTD é composto por cinco módulos principais: agregação de tráfego de rede, extração de características de comportamento local, extração de características de interação global, aprendizado contrastivo e detecção de anomalias. Utiliza técnicas de agrupamento para formar pares de amostras positivas e negativas, e um gráfico de interação de *hosts* com uma rede neural de atenção de borda dupla (DE-GAT) para capturar características da topologia e atributos do gráfico, realizando a detecção de anomalias em

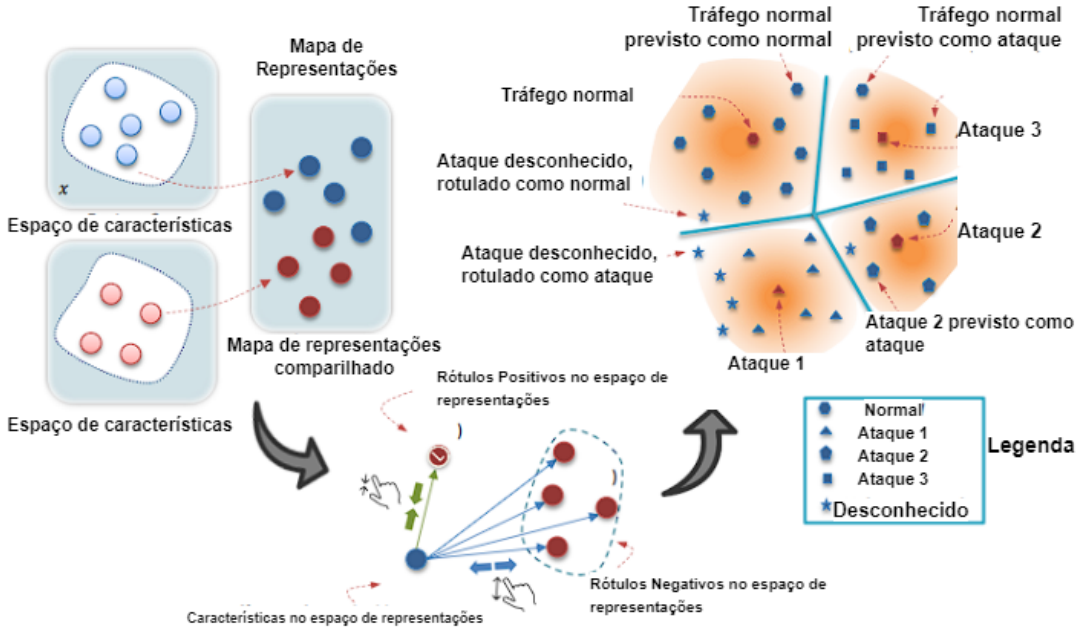

Figura 2.12. As características de amostra e os rótulos são mapeados para o mesmo espaço de incorporação. Cada rótulo cria um hipercone de separação usado para classificar as amostras. Por exemplo, ataques conhecidos e tráfego normal são representados por hipercones separados. Um ataque desconhecido pode ser classificado corretamente como um ataque ou incorretamente como tráfego normal, dependendo de onde cai no espaço de incorporação. Adaptado de [Lopez-Martin et al., 2023].

múltiplas rodadas para melhorar a precisão.

A **detecção de malware** também pode se beneficiar do aprendizado contrastivo. o EVOLIoT [Dib et al., 2022], apresenta uma estrutura de aprendizado contrastivo auto-supervisionado para detectar e caracterizar variantes evolutivas de *malware* em IoT. O método combate o "desvio de conceito" (*concept drift)* e limitações na classificação de *malware* entre famílias. Utilizando representações semânticas de binários de *malware* de IoT, o sistema diferencia amostras evoluídas sem a necessidade de rótulos caros. Avaliações mostram que o sistema melhora a precisão na identificação de variantes e na preservação de informações semânticas em um cenário de *malware* de IoT em rápida evolução. [Yang et al., 2022] apresenta um método para detecção e classificação de *malware* em dispositivos Android baseado em aprendizado contrastivo. Utilizando codificação de características sem *token* e um modelo TextCNN, o sistema extrai características variáveis dos dados de entrada. O treinamento do modelo é realizado com a técnica *Bootstrap Your Own Latent* (BYOL) que não depende de amostras negativas, melhorando a precisão e a robustez do detector.

A solução como a de [Kye et al., 2022] busca detectar anomalias em redes de computadores utilizando aprendizado auto-supervisionado motivado pela necessidade de identificar anomalias extremas que podem causar danos significativos. A solução propõe uma abordagem hierárquica com múltiplos estágios de detecção baseados no nível de anormalidade. Utilizando o espaço oculto do *autoencoder*, a solução é treinada com sinais de auto-supervisão, eliminando a dependência de dados anormais escassos. Avaliada

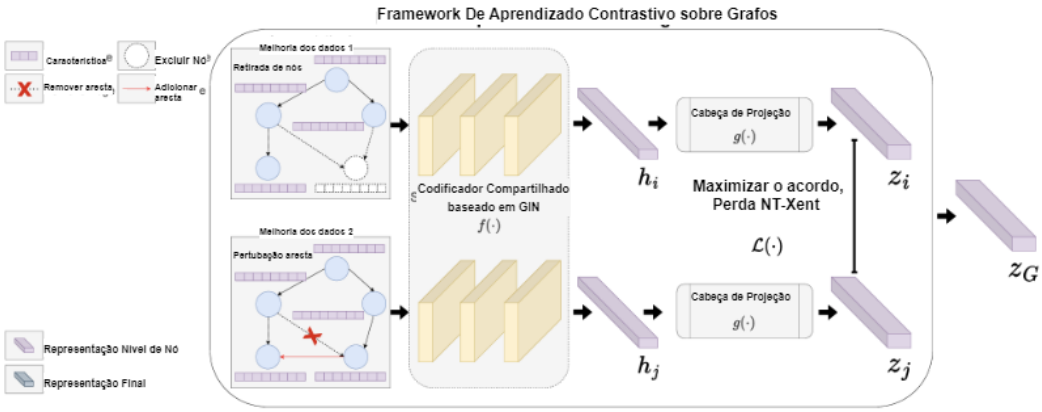

Figura 2.13. Pré-treinamento no aprendizado contrastivo de grafos maximizando a concordância entre visualizações aumentadas do mesmo grafo, utilizando um codificador rede de isomorfismo de grafo (GIN), uma cabeça de projeção não linear, e a função de perda contrastiva NT-Xent para obter representações finais robustas. Adaptado de [Gao et al., 2022].

em conjuntos de dados populares, o método se destaca na detecção eficiente de anomalias extremas em redes de computadores.

Outra aplicação de aprendizado auto-supervisionado trata da **detecção de anomalias em dados de grafo**, uma área crucial para aplicações como segurança de redes e detecção de fraudes. Os métodos tradicionais, inadequados para estruturas complexas não euclidianas, demandam o desenvolvimento de abordagens eficazes. O SL-GAD (Self-Supervised Learning for Graph Anomaly Detection) [Zheng et al., 2021] é proposto para superar essas limitações, adotando aprendizado auto-supervisionado. Gerando duas visualizações de subgrafo do nó alvo, o SL-GAD utiliza reconstrução de atributos generativos e aprendizado contrastivo multi-visão para identificar anomalias.

O SL-GAD se destaca ao construir diferentes subgrafos contextuais (visões) com base em um nó alvo e ao empregar estratégias de aprendizado auto-supervisionado para fazer comparações e obter a pontuação de anomalia para cada nó. Ele utiliza um codificador de rede neural de grafo (GNN) para aprender a representação latente de cada nó a partir das diferentes visões amostradas. Em seguida, os módulos de regressão de atributos generativos e aprendizado contrastivo multi-visão são empregados para explorar as informações disponíveis de forma auto-supervisionada.

Essa abordagem inovadora permite que o SL-GAD capture anomalias em dados de grafo de forma mais eficaz do que os métodos existentes. Ele supera as limitações dos métodos rasos que não conseguem capturar a complexa interdependência dos dados de grafo e dos métodos de autoencoder de grafo que não conseguem explorar totalmente as informações contextuais como sinais de supervisão para detecção eficaz de anomalias.

Uma abordagem para classificação de *malware* em arquivos executáveis usando aprendizado contrastivo em grafos, não supervisão é proposto por [Gao et al., 2022]. O pré-treinamento é realizado maximizando a concordância entre duas visualizações aumentadas do mesmo grafo usando a perda contrastiva no espaço latente. A estrutura consiste em quatro componentes principais: (1) *Aumento de Dados de Grafos*, em que os dados de grafos G são aumentados para gerar dois grafos relacionados $\hat{G}_i$ e $\hat{G}_j$ como pa-

res positivos; (2) *Codificador baseado em rede de isomorfismo de grafo (GIN)*, utilizado para gerar representações vetoriais dos grafos, com três camadas e uma camada oculta de 64 dimensões, em que a função de leitura soma as incorporações de todos os nós para obter as representações iniciais h_i e h_j; (3) *Cabeça de Projeção*, uma transformação não linear que mapeia as representações aumentadas para outro espaço latente, em que a perda contrastiva é calculada, e z_i e z_j são obtidos aplicando um perceptron de duas camadas (MLP); (4) *Função de Perda Contrastiva*, que maximiza a consistência entre pares positivos z_i e z_j e minimiza entre pares negativos, utilizando a perda de entropia cruzada normalizada pela temperatura (NT-Xent) para obter uma representação final do grafo z_G.

2.6.2. Qualidade de Serviço (QoS)

A **recomendação de serviços com base na Qualidade de Serviço (QoS)** tem despertado grande interesse, especialmente no que diz respeito à criação de matrizes de fatoração e à definição de conjuntos de dados para avaliar algoritmos. No entanto, a literatura existente tem se concentrado principalmente na precisão da previsão (MAE/RMSE), negligenciando os tempos de treinamento e invocação, cruciais para a implantação desses algoritmos em dispositivos de borda com recursos limitados. Além disso, ao contrário de cenários estáticos, os fatores de QoS, como tempo de resposta e vazão, são dinâmicos, requerendo um retreinamento frequente dos modelos. Para enfrentar esses desafios, White et al. propuseram uma abordagem que reduz o tempo de treinamento da previsão de QoS utilizando um empilhamento de autocodificadores, adequado para dispositivos de computação em borda, o que facilita a análise de ambientes dinâmicos e influencia a composição efetiva de serviços [White et al., 2019].

O empilhamento de autocodificadores funciona comprimindo os dados de entrada para a camada oculta e, em seguida, decodificando-os. Com essa abordagem, hierarquias úteis são capturadas, levando a um melhor desempenho. O modelo utiliza múltiplas camadas ocultas, treinadas de forma gulosa para obter parâmetros. A primeira camada destaca características de primeira ordem, como bordas, enquanto camadas subsequentes aprendem características de ordem superior. Assim, o autocodificador empilhado estende o modelo, gerando representações mais ricas e adaptadas às necessidades dinâmicas dos sistemas de recomendação de serviços baseados em QoS.

Yin *et al.* abordam a previsão da Qualidade de Serviço (QoS) em serviços de cidades inteligentes [Yin et al., 2023]. O desafio é prever os valores de QoS ausentes, considerando a variabilidade das condições de rede e o estado dos servidores, que aumentam a dimensão e a complexidade dos dados, além de intensificar o problema de esparsidade dos dados. Os autores propõem o CLpred, um método baseado em aprendizado contrastivo que utiliza uma sequência temporal de dados de QoS. O CLpred emprega um codificador *transformer*, que consiste em várias camadas de atenção própria multi-cabeças e redes *feed-forward* posicionais, permitindo modelar a sequência temporal dos dados de QoS. O codificador *transformer* do CLpred é capaz de capturar interações complexas entre usuários e serviços ao longo do tempo, proporcionando representações mais eficientes dos dados. O aprendizado contrastivo no CLpred envolve a criação de amostras positivas e negativas por meio de técnicas de aumento de dados, como recorte, mascaramento e reordenação de sequências de QoS. Essas técnicas ajudam a amplificar os dados e reduzir a esparsidade. O modelo é treinado usando uma função de perda contrastiva, que ajusta as

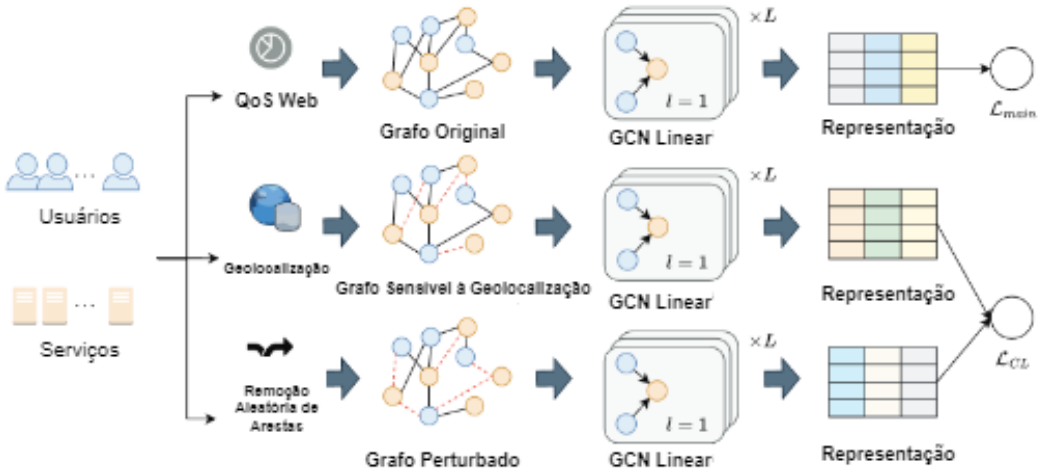

Figura 2.14. O arcabouço *QoS-aware Graph Contrastive Learning* (QAGCL) para recomendação de serviços web. O processo começa com a pré-processamento dos dados de invocação de usuários e serviços para formar um grafo inicial baseado em valores de QoS. Vistas adicionais do grafo são então construídas usando informações de geolocalização e exclusão aleatória de arestas. As incorporações iniciais são processadas por diferentes Redes Convolucionais de Grafos (GCNs) para os grafos aumentados, com uma incorporação usada para recomendação e as outras duas para aprendizado contrastivo. Adaptado de [Choi e Ryu, 2023].

representações dos dados para que amostras positivas, originadas da mesma sequência de usuário, fiquem mais próximas, enquanto amostras negativas, originadas de sequências diferentes, fiquem mais distantes.

Técnicas de aprendizado contrastivo para melhorar a recomendação e a previsão de QoS em serviços web, abordando problemas de esparsidade de dados e cold-start são explorados em [Choi e Ryu, 2023, Zhu et al., 2023]. Choi *et al.* propõem o *QoS-aware graph contrastive learning* (QAGCL), um modelo que utiliza redes convolucionais de grafos e aprendizado contrastivo para integrar informações contextuais, como geolocalização, em grafos aumentados, mostrado na Figura 2.14. Este modelo constrói vistas contextualmente aumentadas para aprender incorporações de usuários e serviços, melhorando a precisão das recomendações de serviços mesmo com interações limitadas. Os experimentos demonstram que o QAGCL supera modelos existentes em termos de precisão de recomendação, especialmente em condições de alta esparsidade de dados. O segundo artigo introduz o BGCL, uma rede bi-subgrafo baseada em aprendizado contrastivo para prever QoS em situações de *cold-start*. O BGCL gera diferentes perspectivas de subgrafos de vizinhança de usuários e serviços a partir de grafos bipartidos esparsos. Em seguida, utiliza aprendizado contrastivo de grafos e mecanismos de agregação de atenção para aprender incorporações de usuários e serviços. Essas incorporações são então alimentados em uma rede perceptora multicamadas para prever valores de QoS. Resultados experimentais mostram que o BGCL supera vários modelos existentes em termos de precisão de previsão, demonstrando eficácia em ambientes de baixa densidade de dados.

A classificação de tráfego de rede criptografado é um desafio crítico para a gestão de redes, qualidade de serviço (QoS) e segurança de redes. Com o crescimento das aplica-

Tabela 2.2. Comparação de trabalhos com foco em detecção de intrusões e provisão de QoS.

Referência	Descrição
[Wang et al., 2022a]	**FeCo - Federated Contrastive Learning Framework:** Solução para melhorar a detecção de intrusões em redes IoT por meio de aprendizado contrastivo federado, coordenando dispositivos IoT para aprender modelos de detecção de intrusão de forma conjunta.
[Lopez-Martin et al., 2023]	**Aprendizado Contrastivo sobre Características de Fourier Aleatórias:** Técnica que mapeia as características da rede e os rótulos para um espaço comum para medir a similaridade e realizar a classificação, especialmente otimizada para identificar ataques desconhecidos.
[Yue et al., 2022]	Técnica de aprendizado contrastivo utilizando mascaramento aleatório de sequências de pacotes de rede para criar tarefas contrastivas, melhorando a precisão e a taxa de detecção de intrusões em ambientes complexos de rede.
[Li et al., 2024a]	Aprendizado contrastivo para melhorar a distinção entre tráfego benigno e malicioso no espaço de representação, resultando em maior precisão na detecção de ataques desconhecidos.
[Zhang et al., 2024]	**AOC-IDS:** Sistema de detecção de intrusões em tempo real com um módulo de detecção de anomalias (ADM) e um framework em tempo real que permite adaptação contínua usando um autoencoder com uma função de perda contrastiva personalizada.
[Han et al., 2024]	**ContraMTD:** Método não supervisionado para detecção de tráfego malicioso, extraindo características de comportamento local e interação global do tráfego normal usando aprendizado contrastivo.
[Dib et al., 2022]	**EVOLIoT:** Estrutura de aprendizado contrastivo auto-supervisionado para detectar e caracterizar variantes evolutivas de malware em IoT, utilizando representações semânticas de binários de malware.
[Yang et al., 2022]	Método para detecção e classificação de malware em dispositivos Android baseado em aprendizado contrastivo, utilizando codificação de características sem token e um modelo TextCNN.
[Kye et al., 2022]	Abordagem hierárquica para detecção de anomalias em redes de computadores utilizando aprendizado auto-supervisionado com múltiplos estágios de detecção baseados no nível de anormalidade.
[Zheng et al., 2021]	**SL-GAD:** Aprendizado auto-supervisionado para detecção de anomalias em dados de grafo, utilizando reconstrução de atributos generativos e aprendizado contrastivo multi-visão para identificar anomalias.
[Gao et al., 2022]	Abordagem para classificação de malware em arquivos executáveis usando aprendizado contrastivo em grafos não supervisionado, maximizando a concordância entre visualizações aumentadas do mesmo grafo.
[White et al., 2019]	Redução do tempo de treinamento da previsão de QoS usando um empilhamento de autocodificadores, adequado para dispositivos de computação em borda.
[Yin et al., 2023]	**CLpred:** Método baseado em aprendizado contrastivo para previsão de QoS em serviços de cidades inteligentes, utilizando um codificador transformer para modelar a sequência temporal dos dados de QoS.
[Choi e Ryu, 2023]	**QAGCL:** Modelo de aprendizado contrastivo de grafos para recomendação de serviços web, integrando informações contextuais como geolocalização.
[Zhu et al., 2023]	**BGCL:** Rede bi-subgrafo baseada em aprendizado contrastivo para prever QoS em situações de cold-start, utilizando mecanismos de agregação de atenção para aprender embeddings de usuários e serviços.
[Tian et al., 2022]	Método de identificação de tráfego criptografado baseado em aprendizado contrastivo, utilizando um modelo pré-treinado e técnicas de clusterização para adicionar pseudo-rótulos.
[Wang et al., 2022b]	Abordagem utilizando Redes Neurais de Grafo (GNN) e Redes de Ponteiro (PN) para a composição de serviços cientes da Qualidade de Serviço (QoS), utilizando um algoritmo de otimização baseado no comportamento de baleias para ajustar a solução inicial.

ções e protocolos criptografados, métodos tradicionais de detecção tornaram-se ineficazes devido à perda de informações semânticas e à dificuldade na extração de características. [Tian et al., 2022] propõe um método de identificação de tráfego criptografado baseado em aprendizado contrastivo. Este método utiliza um modelo pré-treinado com aprendizado contrastivo supervisionado e expande o conjunto de dados rotulados por meio de técnicas de clusterização para adicionar pseudo-rótulos. O algoritmo é baseado em três componentes principais: i) Seleção de Granularidade: utiliza sessões, que são coleções de fluxos definidos pelas cinco-tupla, IP de origem, IP de destino, porta de origem, porta de destino e protocolos de nível de transporte; ii) Processamento de Dados Não Rotulados: usa Análise de Componentes Principais (PCA) para reduzir a dimensionalidade dos dados de tráfego e técnicas de agrupamento para expandir o conjunto de dados rotulados com pseudo-rótulos; iii) Identificação de Tráfego Criptografado: implementa aprendizado contrastivo supervisionado com uma rede neural ResNet para extração e projeção de características.

Wang *et al.* propõem uma abordagem utilizando Redes Neurais de Grafo (GNN) e Redes de Ponteiro (PN), baseado em aprendizado por reforço, para a composição de serviços cientes da Qualidade de Serviço (QoS) [Wang et al., 2022b]. Primeiramente, os dados de tarefas e serviços são estruturados como grafos, permitindo que a GNN extraia correlações subjacentes e preveja a probabilidade de uso de cada serviço. Com base nesses serviços de alta probabilidade, a rede de ponteiro, frequentemente utilizada para problemas de otimização combinatória, é empregada para construir a solução inicial de serviços. Adicionalmente, para melhorar a capacidade de generalização da rede, uma camada extra é adicionada à PN. Finalmente, um algoritmo de otimização baseado no comportamento de baleias é utilizado para ajustar a solução inicial, incorporando serviços raramente usados.

Detecção de anomalias em dados de séries temporais multivariadas (MTS), com foco especial em sua aplicação em cibersegurança para detecção de ataques desconhecidos. [González et al., 2023] apresentam o DC-VAE, uma abordagem recente que utiliza Variational Auto Encoders (VAEs) e Dilated Convolutional Neural Networks (DCNNs) para modelar dados MTS complexos e de alta dimensão. No entanto, os autores reconhecem que a detecção de anomalias usando VAEs pode resultar em degradação de desempenho e até esquecimento catastrófico quando treinados em medidas de rede dinâmicas e em evolução, especialmente em casos de mudanças no conceito dos dados. Portanto, eles propõem uma extensão do DC-VAE para um cenário de aprendizado contínuo, aproveitando as propriedades da inteligência artificial generativa dos modelos subjacentes para lidar com dados em constante evolução.

2.7. Tendências e desafios de pesquisa

A aplicação de técnicas de aprendizado auto-supervisionado (*Self-Supervised Learning* - SSL) em redes de computadores visa resolver desafios complexos, como a classificação de tráfego, a detecção de intrusões e a otimização do desempenho. Para maximizar a eficácia dessas técnicas, é crucial abordar questões específicas como a rotulagem dos dados, a evolução dos métodos de aprendizado e a adaptação dos modelos à natureza dinâmica dos dados de rede. Além disso, alguns desafios estão relacionados a outros paradigmas de aprendizado de máquina, como a falta de explicabilidade e a ausência de

fundamentação teórica. Esta seção explora diversas abordagens e estratégias emergentes no campo, destacando suas contribuições e desafios em redes de computadores.

Rotulagem dos dados: O desafio crucial na implantação de mecanismos de aprendizado de máquina em aplicações de redes de computadores é a rotulagem dos dados. O tráfego de rede, frequentemente gerado a altas taxas, contrasta drasticamente com a taxa de rotulagem realizada por especialistas humanos. A discrepância se manifesta em uma escala de poucos fluxos de rede por evento de rotulagem frente à velocidade de linha de transmissão de dados em enlaces monitorados. Diante desse desafio, o domínio de aplicações de redes busca soluções para tornar a rotulagem de dados mais eficiente, especialmente considerando o aumento exponencial do volume de dados. O aprendizado auto-supervisionado na área de redes está se desenvolvendo para criar representações mais eficientes e significativas dos dados, permitindo uma compreensão mais profunda dos padrões de tráfego.

Evolução do aprendizado contrastivo: Uma tendência notável nesse campo é o crescimento do paradigma contrastivo. Nesse método, são utilizados pares de dados, em que cada par consiste em uma instância de dados e uma versão modificada ou distorcida dessa instância. O objetivo é ensinar ao modelo a distinguir entre instâncias semelhantes e diferentes. Para fazer isso, durante o treinamento, o modelo é incentivado a aproximar instâncias semelhantes no espaço latente, enquanto afasta instâncias diferentes. Isso é alcançado por meio de técnicas que minimizam a distância entre instâncias similares e maximizam a distância entre instâncias diferentes no espaço de representação.

Essa abordagem contrastiva tem se mostrado muito eficaz na aprendizagem de representações de alta qualidade, especialmente em conjuntos de dados grandes e complexos. Ao aprender a identificar padrões significativos nos dados através da comparação de instâncias, os modelos construídos com base nesse paradigma conseguem capturar nuances sutis nos dados, resultando em representações mais ricas e informativas. Isso, por sua vez, leva a melhorias significativas no desempenho de tarefas relacionadas à análise de tráfego de rede, como classificação de aplicativos, detecção de intrusões e previsão de falhas.

Transferência de conhecimento: A transferência de conhecimento é uma estratégia fundamental no campo do aprendizado de máquina, que visa aproveitar o conhecimento adquirido em um domínio específico para melhorar o desempenho em tarefas relacionadas ou diferentes. Essa abordagem é particularmente útil quando há uma escassez de dados rotulados em um domínio-alvo, mas abundância de dados em um domínio-fonte relacionado. Uma das formas mais comuns de transferência de conhecimento é a técnica de pré-treinamento de modelos em grandes conjuntos de dados genéricos, seguida pelo ajuste fino (*fine-tuning*) em conjuntos de dados específicos da tarefa. Essa abordagem permite que modelos pré-treinados capturem padrões gerais nos dados durante a fase de pré-treinamento e, em seguida, ajustem-se aos padrões específicos do novo domínio durante o ajuste fino. Além disso, a aplicação multidomínio da transferência de conhecimento destaca-se como uma tendência importante. Nesse contexto, modelos pré-treinados são utilizados em uma variedade de domínios, explorando o aprendizado auto-supervisionado em diversas áreas. Isso significa que os modelos podem ser treinados em conjuntos de dados que abrangem diferentes aspectos, como segurança cibernética, otimização de de-

sempenho de redes e detecção de anomalias. Essa abordagem multidomínio permite que os modelos adquiram uma compreensão mais abrangente dos padrões e características dos dados, o que pode levar a um melhor desempenho em uma ampla gama de tarefas e cenários. A transferência de conhecimento também pode ocorrer através do treinamento de modelos auto-supervisionados para aprender representações significativas de dados, que são então utilizados para outras tarefas distintas, como a classificação de tráfego com base nas representações aprendidas.

Coleta, armazenamento de dados pré-processados e adaptação de métodos de aprendizado de máquina à natureza dos dados: O armazenamento de quantidades infinitas de dados é inviável, e a obtenção de dados na natureza muitas vezes implica custos de tempo devido a limitações de largura de banda ou velocidade do sensor. Isso torna o treinamento baseado em épocas impraticável e uma implementação ingênua de abordagens SSL convencionais, utilizando cada amostra apenas uma vez, resultaria em aprendizes ineficientes. Uma solução é utilizar buffers de replay para separar a aquisição de dados do pipeline de treinamento. Uma questão importante é avaliar a eficácia desses mecanismos de replay em permitir que as representações continuem a melhorar enquanto os dados estão sendo coletados.

Aprendizado continuado e não-estacionariedade dos dados: Os dados do mundo real são não-estacionários, apresentando variações temporais significativas. Por exemplo, durante a Copa do Mundo, há um aumento no número de imagens relacionadas ao futebol. Além disso, robôs explorando ambientes internos encontram distribuições semânticas temporalmente agrupadas. Um sistema inteligente de aprendizado contínuo deve ser capaz de assimilar novos conceitos sem esquecer os antigos em meio a essas mudanças. No entanto, abordagens convencionais de aprendizado contrastivo podem se ajustar excessivamente à distribuição atual, levando ao esquecimento de informações anteriores. Portanto, a questão central é como projetar métodos SSL capazes de aprender efetivamente em ambientes não-estacionários.

Necessidade de novas técnicas de *Data Augmentation*: Avanços recentes na aprendizagem de representações visuais são atribuídos a estratégias de *data augmentation*, como redimensionamento, rotação e coloração. No entanto, aplicar essas técnicas diretamente a dados de grafos é desafiador devido à sua natureza não euclidiana. As estratégias de aumento de dados em grafos geralmente envolvem adicionar ou remover nós e arestas. Para melhorar o aprendizado auto-supervisionado em grafos, é importante projetar estratégias de aumento mais eficientes, seguindo diretrizes específicas e garantindo que sejam aplicáveis, adaptáveis, eficientes e dinâmicas.

Falta de explicabilidade: Embora os métodos de SSL em grafos tenham alcançado bons resultados em diversas tarefas, ainda não compreendemos totalmente o que eles aprendem nas tarefas de pretexto auto-supervisionadas. É importante entender se esses métodos capturam padrões de características, estruturas significativas ou relações entre características e estruturas. Além disso, é necessário determinar se esse aprendizado é explícito ou implícito e se é possível encontrar interpretações claras nos dados de entrada. Essas questões são cruciais para entender o comportamento do modelo, mas estão ausentes na maioria dos trabalhos atuais de SSL em redes. Portanto, é necessário explorar a interpretabilidade desses métodos e analisar profundamente o comportamento do mo-

delo para melhorar sua generalização e robustez em tarefas relacionadas à segurança ou privacidade.

Falta de fundamentação teórica: Apesar do sucesso do SSL em grafos em várias tarefas, a maioria dos métodos existentes baseia-se na intuição, carecendo de fundamentação teórica sólida. Isso resulta em limitações de desempenho e explicabilidade. Construir uma base teórica sólida para o SSL em redes de computadores, minimizando a lacuna entre teoria e prática, é uma direção promissora. Por exemplo, é importante investigar se a maximização da informação mútua é o único método para o aprendizado contrastivo em grafos. Além disso, embora tenhamos introduzido objetivos contrastivos alternativos, como margem de triplo e perda de quadruplo, a conexão teórica entre essas abordagens e a informação mútua ainda precisa ser explorada mais profundamente.

Margem de pré-treinamento: A estratégia comum de treinamento em aprendizado auto-supervisionado (SSL) de grafos envolve o pré-treinamento com tarefas auto-supervisionadas e, em seguida, o uso do modelo pré-treinado para tarefas específicas, seja ajustando os pesos ou mantendo-os congelados. No entanto, a transferência do conhecimento pré-treinado para as tarefas secundárias continua sendo um desafio latente. Embora inúmeras estratégias tenham sido propostas para resolver esse problema nos domínios de visão computacional e processamento de linguagem natural, aplicá-las diretamente a grafos e redes é desafiador devido à sua estrutura não euclidiana inerente. Portanto, é essencial projetar técnicas específicas para grafos que minimizem a diferença entre o pré-treinamento e as tarefas secundárias.

2.8. Descrição da prática da aplicação de aprendizado auto-supervisionado

Esse capítulo se complementa de uma prática da utilização de algoritmos de aprendizado auto-supervisionado para a criação de um sistema de detecção de intrusão. Os participantes do curso são convidados a programar algoritmos de aprendizado de máquina auto-supervisionado, focando a análise de tráfego de redes utilizando conjuntos de dados como NSL-KDD[1] e CICIDS 2017[2]. O roteiro[3] se inicia com a configuração do ambiente de desenvolvimento, em que os participantes instalarão e configurarão todas as ferramentas e bibliotecas necessárias. Em seguida, realizarão uma análise exploratória de dados (*Exploratory Data Analysis* - EDA), em que explorarão as características dos conjuntos de dados, identificarão padrões e anomalias, e realizarão a limpeza dos dados. Posteriormente, serão aplicadas técnicas de *feature engineering* para transformar e preparar os dados de modo adequado para alimentar os modelos de aprendizado auto-supervisionado. Na etapa seguinte, o apresentador explicará detalhadamente o processo de construção do modelo auto-supervisionado, começando pelo pré-treinamento contrastivo, em que os participantes aprenderão a criar representações eficazes dos dados sem a necessidade de rótulos. Após essa etapa, será realizado o refinamento (*fine-tuning*), em que os modelos pré-treinados serão ajustados usando um subconjunto de dados rotulados para melhorar a precisão da detecção de intrusões. Após o treinamento, os participantes interpretarão os resultados, analisando métricas de desempenho como acurácia, precisão, revocação e a

[1]Disponível em `https://www.unb.ca/cic/datasets/nsl.html`.

[2]Disponível em `https://www.unb.ca/cic/datasets/ids-2017.html`.

[3]O roteiro da atividade prática pode ser acessado em `https://github.com/joaovitor-valle/Minicurso-SSL---JAI.git`.

curva ROC. Eles explorarão o ciclo completo, desde a preparação dos dados até a análise de resultados, promovendo uma compreensão sólida dos desafios práticos enfrentados ao lidar com dados reais de tráfego de redes e suas representações. Ao final, espera-se que os participantes tenham adquirido habilidades práticas para a aplicação de algoritmos auto-supervisionados em problemas de redes de computadores, especialmente relacionados à representação de dados de tráfego de redes e à aumentação de dados rotulados.

2.9. Considerações finais

A evolução contínua das ameaças cibernéticas exige que os (*Intrusion Detection Systems* - IDS) se adaptem constantemente para enfrentar novas e sofisticadas formas de ataques. A integração de técnicas avançadas, como inteligência artificial, é cada vez mais comum para melhorar a precisão na detecção e reduzir falsos positivos. A evolução dos algoritmos de aprendizado por grafos e o surgimento de modelos mais complexos que envolvem aprendizado auto-supervisionado têm o potencial de transformar a forma como as máquinas aprendem e representam informações. Esse capítulo abordou o uso do aprendizado auto-supervisionado em aplicações dinâmicas de redes de computadores. O uso de técnicas avançadas de aprendizado de máquina, como o aprendizado auto-supervisionado, oferece uma solução promissora ao enfrentar a escassez de dados rotulados. O capítulo apresentou uma revisão detalhada e atualizada do estado da arte da aplicação do aprendizado auto-supervisionado à criação de sistemas de detecção de intrusões, incluindo os principais modelos e arcabouços, bem como as vantagens e desvantagens de cada abordagem. O capítulo disponibiliza ainda uma atividade prática disponível no repositório `https://github.com/joaovitor-valle/Minicurso-SSL---JAI.git`.

O potencial do aprendizado auto-supervisionado para transformar a segurança e a eficiência das redes de computadores é imenso. A contínua exploração dessas técnicas promete melhorias significativas em termos de segurança, além de expandir os horizontes do aprendizado de máquina em cenários complexos e dinâmicos. A pesquisa nessa área é fortemente incentivada, dada a sua capacidade de enfrentar desafios críticos e abrir novas fronteiras tecnológicas. Contudo, é crucial abordar os problemas e superar as limitações existentes para avançar no uso de soluções de aprendizado auto-supervisionado na área de redes de computadores. Algumas das direções futuras incluem a exploração de novas abordagens, como a análise temporal do tráfego de rede, e a aplicação de outras técnicas como transferência de conhecimento para tornar a predição e o treinamento de modelos mais eficientes de forma distribuída.

O capítulo discutiu ainda diversos tópicos fundamentais e tendência de pesquisas para o aprendizado auto-supervisionado, incluindo a rotulagem de dados, a evolução do aprendizado contrastivo, a transferência de conhecimento, a coleta e armazenamento de dados pré-processados, e a adaptação dos métodos de aprendizado de máquina à natureza dos dados de rede. Adicionalmente, abordaram-se os desafios do aprendizado contínuo em cenários de dados não-estacionários, a necessidade de novas técnicas de *data augmentation* e as lacunas na explicabilidade e fundamentação teórica dos modelos. Também foi destacada a importância da margem de pré-treinamento como uma estratégia crítica para o sucesso das aplicações de aprendizado auto-supervisionado em redes e grafos.

Os principais desafios atuais para área de pesquisa incluem a discrepância entre

a alta taxa de geração de tráfego de rede e a taxa de rotulagem manual, a complexidade de adaptar métodos de aprendizado a dados dinâmicos e não-euclidianos, e a necessidade de melhorar a explicabilidade e fundamentação teórica dos modelos. No entanto, essas dificuldades também representam oportunidades significativas de pesquisa. A pesquisa contínua em métodos auto-supervisionados, o desenvolvimento de técnicas de *feature engineering* e a exploração de novos paradigmas de transferência de conhecimento são áreas promissoras que podem levar a avanços substanciais em aplicações dinâmicas de redes com a detecção de intrusões e otimização de redes com amparo de técnicas de aprendizado auto-supervisionado generativo e contrastivo.

Agradecimentos

Este capítulo foi realizado com recursos do CNPq, FAPERJ, CAPES, RNP e INCT ICONIOT.

Referências

[Abusitta et al., 2023] Abusitta, A., de Carvalho, G. H., Wahab, O. A., Halabi, T., Fung, B. C. e Mamoori, S. A. (2023). Deep learning-enabled anomaly detection for iot systems. *Internet of Things*, 21:100656.

[Bachman et al., 2019] Bachman, P., Hjelm, R. D. e Buchwalter, W. (2019). Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32.

[Barbosa et al., 2024] Barbosa, G. N. N., Andreoni, M. e Mattos, D. M. F. (2024). Optimizing feature selection in intrusion detection systems: Pareto dominance set approaches with mutual information and linear correlation. *Ad Hoc Networks*, 159:103485.

[Barbosa et al., 2021a] Barbosa, G. N. N., Andreoni Lopez, M., Medeiros, D. S. V. e Mattos, D. M. F. (2021a). An entropy-based hybrid mechanism for large-scale wireless network traffic prediction. Em *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, p. 1–6.

[Barbosa et al., 2021b] Barbosa, G. N. N., Bezerra, G. M. G., de Medeiros, D. S. V., Andreoni Lopez, M. e Mattos, D. M. F. (2021b). Segurança em Redes 5G: Oportunidades e Desafios em Detecção de Anomalias e Predição de Tráfego baseadas em Aprendizado de Máquina. Em *Minicursos do XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, p. 145–189.

[Bardes et al., 2022] Bardes, A., Ponce, J. e LeCun, Y. (2022). Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *ICLR, Vicreg*, 1:2.

[Bochie et al., 2020] Bochie, K., da Silva Gilbert, M., Gantert, L., Barbosa, M. d. S. M., de Medeiros, D. S. V. e Campista, M. E. M. (2020). Aprendizado profundo em redes desafiadoras: Conceitos e aplicações. *Sociedade Brasileira de Computação*.

[Caron et al., 2020] Caron, M., Misra, I., Mairal, J., Goyal, P. e Bojanowski, P. (2020). Unsupervised learning of visual features by contrasting cluster assignments. Em *European Conference on Computer Vision*, p. 3–19.

[Caron et al., 2018] Caron, M., Sun, R. e Schölkopf, B. (2018). Counterfactuals uncover the modular structure of deep generative models. *arXiv preprint arXiv:1812.03253*.

[Caron et al., 2021] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P. e Joulin, A. (2021). Emerging properties in self-supervised vision transformers. Em *Proceedings of the IEEE/CVF international conference on computer vision*, p. 9650–9660.

[Chen et al., 2020a] Chen, T., He, X., Fan, Y., Zhang, Y. e Xie, J. (2020a). Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*.

[Chen et al., 2020b] Chen, T., Kornblith, S., Norouzi, M. e Hinton, G. (2020b). A simple framework for contrastive learning of visual representations. Em *International conference on machine learning*, p. 1597–1607. PMLR.

[Chen e He, 2021] Chen, X. e He, K. (2021). Exploring simple siamese representation learning. Em *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, p. 15750–15758.

[Choi e Ryu, 2023] Choi, J. e Ryu, D. (2023). Qos-aware graph contrastive learning for web service recommendation. Em *2023 30th Asia-Pacific Software Engineering Conference (APSEC)*, p. 171–180. IEEE.

[Creswell et al., 2018] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. e Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65.

[Dai et al., 2019] Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V. e Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

[de Oliveira et al., 2021] de Oliveira, N. R., Pisa, P. S., Andreoni Lopez, M., de Medeiros, D. S. V. e Mattos, D. M. F. (2021). Identifying fake news on social networks based on natural language processing: Trends and challenges. *Information*, 12(1).

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. e Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. Em *CVPR09*.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K. e Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Dib et al., 2022] Dib, M., Torabi, S., Bou-Harb, E., Bouguila, N. e Assi, C. (2022). Evoliot: A self-supervised contrastive learning framework for detecting and characterizing evolving iot malware variants. Em *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, p. 452–466.

[Doersch et al., 2015] Doersch, C., Gupta, A. e Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. Em *Proceedings of the IEEE international conference on computer vision*, p. 1422–1430.

[Ermolov et al., 2021] Ermolov, A., Siarohin, A., Sangineto, E. e Sebe, N. (2021). Whitening for self-supervised representation learning. Em *International conference on machine learning*, p. 3015–3024. PMLR.

[Gao et al., 2022] Gao, Y., Hasegawa, H., Yamaguchi, Y. e Shimada, H. (2022). Unsupervised graph contrastive learning with data augmentation for malware classification. Em *Proc. 16th International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2022), IARIA*, p. 41–47.

[Germain et al., 2015] Germain, M., Gregor, K., Murray, I. e Larochelle, H. (2015). Made: Masked autoencoder for distribution estimation. *International Conference on Machine Learning*, p. 881–889.

[Gidaris et al., 2018] Gidaris, S., Singh, P. e Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

[González et al., 2023] González, G. G., Casas, P. e Fernández, A. (2023). Fake it till you detect it: Continual anomaly detection in multivariate time-series using generative ai. Em *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, p. 558–566. IEEE.

[Goyal et al., 2021] Goyal, P., Caron, M., Lefaudeux, B., Xu, M., Wang, P., Pai, V., Singh, M., Liptchinsky, V., Misra, I., Joulin, A. et al. (2021). Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*.

[Grill et al., 2020] Grill, J.-B., Strub, F., Altche, F., Tallec, C. L. e Richemond, P. H. (2020). Bootstrap your own latent: A new approach to self-supervised learning. Em *Advances in Neural Information Processing Systems*, p. 33–44.

[Gutmann e Hyvärinen, 2010] Gutmann, M. e Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. Em *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, p. 297–304. JMLR Workshop and Conference Proceedings.

[Han et al., 2024] Han, X., Cui, S., Qin, J., Liu, S., Jiang, B., Dong, C., Lu, Z. e Liu, B. (2024). Contramtd: An unsupervised malicious network traffic detection method based on contrastive learning. Em *Proceedings of the ACM on Web Conference 2024*, p. 1680–1689.

[Hassani e Khasahmadi, 2020] Hassani, K. e Khasahmadi, A. H. (2020). Contrastive multi-view representation learning on graphs. Em *International conference on machine learning*, p. 4116–4126. PMLR.

[He et al., 2020a] He, K., Fan, H., Wu, Y., Xie, S. e Girshick, R. (2020a). Momentum contrast for unsupervised visual representation learning. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 9729–9738.

[He et al., 2020b] He, K., Fan, H., Wu, Y., Xie, S. e Girshick, R. (2020b). Momentum contrast for unsupervised visual representation learning. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 9729–973.

[Jaiswal et al., 2021] Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D. e Makedon, F. (2021). A survey on contrastive self-supervised learning. *Technologies*, 9(1).

[Jing e Tian, 2020] Jing, L. e Tian, Y. (2020). Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058.

[Kim et al., 2018] Kim, D., Cho, D., Yoo, D. e Kweon, I. S. (2018). Learning image representations by completing damaged jigsaw puzzles. Em *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, p. 793–802. IEEE.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I. e Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

[Kye et al., 2022] Kye, H., Kim, M. e Kwon, M. (2022). Hierarchical detection of network anomalies: A self-supervised learning approach. *IEEE Signal Processing Letters*, 29:1908–1912.

[Lan et al., 2019] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. e Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

[Le-Khac et al., 2020] Le-Khac, P. H., Healy, G. e Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934.

[Lee et al., 2023] Lee, P., Bubeck, S. e Petro, J. (2023). Benefits, limits, and risks of gpt-4 as an ai chatbot for medicine.

[Li et al., 2021] Li, C., Yang, J., Zhang, P., Gao, M., Xiao, B., Dai, X., Yuan, L. e Gao, J. (2021). Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*.

[Li et al., 2024a] Li, L., Lu, Y., Yang, G. e Yan, X. (2024a). End-to-end network intrusion detection based on contrastive learning. *Sensors*, 24(7).

[Li et al., 2024b] Li, Z., Xia, L., Xu, Y. e Huang, C. (2024b). Gpt-st: Generative pre-training of spatio-temporal graph neural networks. *Advances in Neural Information Processing Systems*, 36.

[Liu et al., 2019a] Liu, H., Li, S., Li, H., Li, X., Gao, J. e Ji, S. (2019a). Graphaf: A flow-based autoregressive model for molecular graph generation. Em *Proceedings of the 36th International Conference on Machine Learning*, p. 3872–3881.

[Liu et al., 2023] Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J. e Tang, J. (2023). Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876.

[Liu et al., 2019b] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. e Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[Lopez-Martin et al., 2023] Lopez-Martin, M., Sanchez-Esguevillas, A., Arribas, J. I. e Carro, B. (2023). Contrastive learning over random fourier features for iot network intrusion detection. *IEEE Internet of Things Journal*, 10(10):8505–8513.

[McMahan et al., 2017] McMahan, B., Moore, E., Ramage, D., Hampson, S. e y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Em *Artificial intelligence and statistics*, p. 1273–1282. PMLR.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. e Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

[Misra e Maaten, 2020] Misra, I. e Maaten, L. v. d. (2020). Self-supervised learning of pretext-invariant representations. Em *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, p. 6707–6717.

[Noroozi e Favaro, 2016] Noroozi, M. e Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. Em *European conference on computer vision*, p. 69–84. Springer.

[Oord et al., 2016a] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. e Kavukcuoglu, K. (2016a). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

[Oord et al., 2016b] Oord, A. v. d., Kalchbrenner, N. e Kavukcuoglu, K. (2016b). Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*.

[Oord et al., 2018] Oord, A. v. d., Li, Y. e Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[Peng et al., 2021] Peng, J., Liu, D., Xu, S. e Li, H. (2021). Generating diverse structure for image inpainting with hierarchical vq-vae. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 10775–10784.

[Perozzi et al., 2014] Perozzi, B., Al-Rfou, R. e Skiena, S. (2014). Deepwalk: Online learning of social representations. Em *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 701–710. ACM.

[Popova et al., 2019] Popova, M., Shvets, M., Oliva, J. e Isayev, O. (2019). Molecularrnn: Generating realistic molecular graphs with optimized properties. *arXiv preprint arXiv:1905.13372*.

[Ragab et al., 2022] Ragab, M., Eldele, E., Chen, Z., Wu, M., Kwoh, C.-K. e Li, X. (2022). Self-supervised autoregressive domain adaptation for time series data. *IEEE Transactions on Neural Networks and Learning Systems*.

[Razavi et al., 2019] Razavi, A., Van den Oord, A. e Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32.

[Stafford, 2020] Stafford, V. (2020). Zero trust architecture. *NIST special publication*, 800:207.

[Tang et al., 2015] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J. e Mei, Q. (2015). Line: Large-scale information network embedding. Em *Proceedings of the 24th international conference on world wide web*, p. 1067–1077.

[Thisanke et al., 2023] Thisanke, H., Deshan, C., Chamith, K., Seneviratne, S., Vidanaarachchi, R. e Herath, D. (2023). Semantic segmentation using vision transformers: A survey. *Engineering Applications of Artificial Intelligence*, 126:106669.

[Tian et al., 2022] Tian, S., Gao, Y., Yuan, G., Zhang, R., Zhao, J. e Zhang, S. (2022). An encrypted traffic classification method based on contrastive learning. Em *Proceedings of the 8th International Conference on Communication and Information Processing*, p. 101–105.

[Tian et al., 2020] Tian, Y., Krishnan, D. e Isola, P. (2020). Contrastive multiview coding. Em *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, p. 776–794. Springer.

[Torbarina et al., 2024] Torbarina, L., Ferkovic, T., Roguski, L., Mihelcic, V., Sarlija, B. e Kraljevic, Z. (2024). Challenges and opportunities of using transformer-based multi-task learning in nlp through ml lifecycle: A position paper. *Natural Language Processing Journal*, 7:100076.

[Van den Oord et al., 2016] Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A. et al. (2016). Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. e Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Wang et al., 2022a] Wang, N., Chen, Y., Hu, Y., Lou, W. e Hou, Y. T. (2022a). Feco: Boosting intrusion detection capability in iot networks via contrastive learning. Em *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, p. 1409–1418. IEEE.

[Wang et al., 2022b] Wang, X., Xu, H., Wang, X., Xu, X. e Wang, Z. (2022b). A graph neural network and pointer network-based approach for qos-aware service composition. *IEEE Transactions on Services Computing*.

[Wei et al., 2019] Wei, C., Xie, L., Ren, X., Xia, Y., Su, C., Liu, J., Tian, Q. e Yuille, A. L. (2019). Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 1910–1919.

[Wen et al., 2020] Wen, L., Li, X. e Gao, L. (2020). A transfer convolutional neural network for fault diagnosis based on resnet-50. *Neural Computing and Applications*, 32(10):6111–6124.

[White et al., 2019] White, G., Palade, A., Cabrera, C. e Clarke, S. (2019). Autoencoders for qos prediction at the edge. Em *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom*, p. 1–9.

[Wu et al., 2023] Wu, L., Lin, H., Tan, C., Gao, Z. e Li, S. Z. (2023). Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):4216–4235.

[Yang et al., 2022] Yang, S., Wang, Y., Xu, H., Xu, F. e Chen, M. (2022). An android malware detection and classification approach based on contrastive lerning. *Computers & Security*, 123:102915.

[Yang et al., 2019] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R. e Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

[Yin et al., 2023] Yin, Y., Di, Q., Wan, J. e Liang, T. (2023). Time-aware smart city services based on qos prediction: A contrastive learning approach. *IEEE Internet of Things Journal*.

[You et al., 2018a] You, J., Liu, B., Ying, Z., Pande, V. e Leskovec, J. (2018a). Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31.

[You et al., 2018b] You, J., Ying, R., Ren, X., Hamilton, W. e Leskovec, J. (2018b). Graphrnn: Generating realistic graphs with deep auto-regressive models. Em *International Conference on Machine Learning*, p. 5708–5717.

[You et al., 2017] You, Y., Gitman, I. e Ginsburg, B. (2017). Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*.

[Yue et al., 2022] Yue, Y., Chen, X., Han, Z., Zeng, X. e Zhu, Y. (2022). Contrastive learning enhanced intrusion detection. *IEEE Transactions on Network and Service Management*, 19(4):4232–4247.

[Zafar et al., 2022] Zafar, S., Lv, Z., Zaydi, N. H., Ibrar, M. e Hu, X. (2022). Dsmlb: Dynamic switch-migration based load balancing for software-defined iot network. *Computer Networks*, 214:109145.

[Zbontar et al., 2021] Zbontar, J., Jing, L., Misra, I., LeCun, Y. e Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. Em *International conference on machine learning*, p. 12310–12320. PMLR.

[Zhang et al., 2024] Zhang, X., Zhao, R., Jiang, Z., Sun, Z., Ding, Y., Ngai, E. C. e Yang, S.-H. (2024). Aoc-ids: Autonomous online framework with contrastive learning for intrusion detection. *arXiv preprint arXiv:2402.01807*.

[Zhang e Zhu, 2023] Zhang, X. e Zhu, Q. (2023). Ai-enabled network-functions virtualization and software-defined architectures for customized statistical qos over 6g massive mimo mobile wireless networks. *IEEE Network*, 37(2):30–37.

[Zhao et al., 2023] Zhao, W., Xu, G., Cui, Z., Luo, S., Long, C. e Zhang, T. (2023). Deep graph structural infomax. Em *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, p. 4920–4928.

[Zheng et al., 2021] Zheng, Y., Jin, M., Liu, Y., Chi, L., Phan, K. T. e Chen, Y.-P. P. (2021). Generative and contrastive self-supervised learning for graph anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*.

[Zhu et al., 2023] Zhu, J., Li, B., Wang, J., Li, D., Liu, Y. e Zhang, Z. (2023). Bgcl: Bi-subgraph network based on graph contrastive learning for cold-start qos prediction. *Knowledge-Based Systems*, 263:110296.

[Zhuang et al., 2019] Zhuang, C., Zhai, A. L. e Yamins, D. (2019). Local aggregation for unsupervised learning of visual embeddings. Em *Proceedings of the IEEE/CVF international conference on computer vision*, p. 6002–6012.

[Zoph et al., 2020] Zoph, B., Cubuk, E. D., Ghiasi, G., Lin, T.-Y., Shlens, J. e Le, Q. V. (2020). Rethinking pre-training and self-training. Em *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 10286–10295.