Big Learning

Yulai Cong* Miaoyun Zhao
Sun Yat-sen University UBTECH

Abstract
Recent advances in big/foundation models reveal a promising path for deep learning, where the roadmap steadily moves from big data to big models to (the newly-introduced) big learning. Specifically, the big learning exhaustively exploits the information inherent in its large-scale complete/incomplete training data, by simultaneously modeling many/all joint/conditional/marginal data distributions across potentially diverse domains, with one universal foundation model. We reveal that big learning (i) underlies most existing foundation models, (ii) is equipped with extraordinary flexibilities for complete/incomplete training data and trustworthy data tasks, (iii) is capable of delivering all joint/conditional/marginal data capabilities with one universal model, and (iv) unifies conventional machine learning paradigms and enables their flexible cooperations, manifested as a universal learning paradigm. Diverse experiments are carried out to validate the effectiveness of the presented big learning.

1 Introduction
AI is undergoing a paradigm shift with the rise of big/foundation models [5, 57], e.g., BERT [47], GPT-3 [7], MAE [21], DALL-E [43, 42], Imagen [45], Stable Diffusion [44], UniDiffuser [3], ChatGPT [39, 40], etc. Foundation models, often based on mask-and-predict pretraining and downstream finetuning, are capable of benefiting from pretraining on broad data at scale and accordingly, demonstrate diverse downstream task capabilities with impressive robustness [47], adaptability [21], and generalization [43]. Therefore, they are rapidly being integrated into real-world AI systems, e.g., BERT into Google search, Codex [8] into GitHub’s Copilot, ChatGPT into Microsoft Bing, etc.

Despite the impressive capabilities and characteristics of foundation models, a unified theoretical framework justifying their great successes remains missing [5, 57], which is believed crucial for their further improvements and is likely a milestone for the foundation model community [48]. We propose the big learning to address that unified-framework challenge. In what follows, we first summarizing the reasons for the successes of existing foundation models and then we sublimate their training objectives into the presented big learning to benefit more from those reasons.

By referring to [5, 57], we attribute the successes of foundation models to the following two properties of their large-scale pretraining.

1. **Data comprehensiveness.** Foundation models are often pretrained with massive data with great diversity. Often collected with minimal human interventions, these pretraining data are likely comprehensively consistent with the “true” data distribution that underlies both training/pretraining and test/finetuning phases, leading to a narrowed phase gap from the data perspective and, therefore, serving as one reason for the generalization and robustness of foundation models.

2. **Task comprehensiveness.** Foundation models are pretrained in a massive multitasking manner on a wealth of data tasks; e.g., both masked language modeling (MLM) and causal LM (CLM)

*Corresponding author: Yulai Cong <yulaicong@gmail.com>.

Preprint. Under review.
leverage one universal model to simultaneously model many conditional data distributions (revealed in Section 3). Such massive-task pretraining shows foundation models comprehensive task experience, which narrows the training-test/pretraining-finetuning gap from the task perspective (it’s likely the downstream task resembles a pretraining one).

Inspired by existing foundation models succeeding from their comprehensive pretraining data and tasks, we propose the big learning that (i) is capable of enhancing both comprehensiveness to the extreme and (ii) unifies the training objectives of most foundation models within one framework. Specifically, the big learning leverages a universal foundation model to simultaneously model many/all joint/conditional/marginal data distributions across potentially diverse domains, manifested as a “big” generative learning task that exhaustively exploits the data information from many/all perspectives.

The big learning naturally emerges from a promising path for deep learning, where the roadmap gradually follows big data, big models, big learning, ···. Specifically, one collects big data to comprehensively represent the underlying data distribution, develops big models to serve as high-capacity information “containers,” relies on big learning to comprehensively and exhaustively convey data information into those containers, and so forth.

Our contributions are summarized as follows.

• We propose the big learning to unify most foundation models within one framework.
• The big learning is capable of delivering all joint/conditional/marginal data capabilities with one universal foundation model. Those data capabilities, in general settings, can manifest as classification, generation, completion/recommendation, etc.
• We empirically demonstrate that big learning (i) is feasible, (ii) delivers great model generalization, and (iii) can serve as a better strategy for finetuning foundation models.

2 Preliminary

Big/Foundation models. Taking shape in natural language processing (NLP), big/foundation models have drastically changed the research and practice of AI [5, 57]. BERT [47] and GPT series [41, 7] significantly accelerate the development of NLP, while models like DALL-Es [43, 42], Stable Diffusion [44], and UniDiffuser [3] effectively promote interdisciplinary research among different research fields, initiating a new revolution of AI-Generated Content (AIGC).

Most existing foundation models are pretrained with (i) masked LM (or masked auto-encoding; like BERT and MAE), (ii) causal/auto-regressive LM (like GPTs and DALL-E), and (iii) permutation LM (like XLNET [56]). We will demonstrate in Section 3 that these pretraining methods are all special cases of the proposed big learning, which, accordingly, serves as a unified theoretical framework that reveals one underlying principle of foundation models.

Transformers and Vision Transformers (ViTs). Based on the flexible self-attention mechanism [50], Transformers have been serving as the de facto model architecture for foundation models. Often Transformers take as input an $L$-length sequence of discrete tokens $x \in \mathbb{Z}^L$ and output the corresponding $D$-dimensional embedding $h \in \mathbb{R}^{L \times D}$, with the self-attention mechanism flexibly customized (among the $L$ locations) to implement masked/causal/permutation LM. ViTs [15] are Transformers modified for modeling continuous image patches. Despite their high model capacity and flexible modeling capabilities, Transformers/ViTs are well-known to be over-parameterized and data/information hungry [42, 20, 52]; we will reveal that those properties of Transformers/ViTs exactly matches the big learning.

Multi-mode training objectives. Two well-known multi-mode training objectives are (i) the cross-entropy loss, often used in maximum likelihood learning with discrete categorical observations, and (ii) the GAN loss [16] for adversarial learning on continuous observations, as detailed below.

1. The cross-entropy loss. Given history observations $x$ and the current word $y$ sampled from the underlying data distribution $q(x, y)$, and a model $p_\theta(y|x)$ modeling the categorical distribution

2 Throughout this paper, generative modeling is used in its broad sense; for example, classification may be viewed as the generative modeling of a label conditioned on its feature.

3 See Appendix D for discussions on pretraining associated with contrastive learning.
of $y$ given $x$, the commonly-used cross-entropy loss is identical to

$$E_{q(x,y)}[-\log p_\theta(y|x)] \propto KL[q(x,y)||p_\theta(y|x)q(x)],$$

(1)

where the optimal $p_\theta^*(y|x) = q(y|x)$. Note the categorical modeling of $p_\theta(y|x)$ can model multiple modes\(^5\) e.g., consider the diverse generation from the GPT-3 [7].

2. The GAN loss. GANs are known for synthesizing highly realistic images with multiple modes [26, 27, 29]. A standard GAN consists of a generator $G_\theta$ and a discriminator $D_\phi$, both of which are trained in an adversarial manner via

$$\min_\theta \max_\phi \mathbb{E}_{q(x)} \log D_\phi(x) + \mathbb{E}_{p_\theta(x)} \log (1 - D_\phi(x)),$$

(2)

where $q(x)$ is the underlying data distribution and $p_\theta(x)$ is the generated distribution with the generative process $x = G_\theta(z), z \sim p(z)$. $p(z)$ is an easy-to-sample distribution, like a normal distribution. With optimal $D_\phi^*$, Eq. (2) minimizes the Jensen-Shannon (JS) divergence $JS[q(x) || p_\theta(x)]$ [16].

To demonstrate the flexibilities of the big learning, we instantiate it within both maximum likelihood and adversarial learning territories (with the multi-mode objectives in (1) and (2), respectively) in Section 3.2 where Transformers/ViTs are employed to construct its universal foundation model.

3 Big Learning

For better introduction of the big learning, we first present its main idea in simplified unsupervised/unimodal settings, where a data sample $\mathbf{X} = (x)$ contains only a feature $x \in \mathbb{R}^{L \times D}$ (with length $L$ and dimension $D$); like $L$ words with $D = 1$ or $L$ flattened image patches with $D$ pixels); we then generalize its scope to the general settings, where a data sample $\mathbf{X} = (y, x)$ contains both feature $x$ and its paired supervision $y \in \mathbb{R}^{L' \times D'^y}$ (e.g., when $L' = D'^y = 1$, $y \in \{1, \cdots, C\}$ may represent a label). In both settings, the big learning can naturally handle “incomplete data,” which are defined as either $x$ missing values along the $L$-dimension or $y$ missing values along the $L'^y$-dimension.

3.1 Unsupervised/Uni-modal Big Learning

Given complete data samples $x \in \mathbb{R}^{L \times D}$ drawn from the underlying data distribution $q(x)$, the mainstream machine learning paradigms concentrate on joint modeling, i.e., to construct a model $p_\theta(x)$ in the joint domain to resemble $q(x)$, or informally $p_\theta(x) \rightarrow q(x)$, leveraging objectives like GANs [6, 24], VAEs [30, 12], Flows [14, 31], diffusion models [23, 46], etc.

However, joint modeling can not take advantage of incomplete data (e.g., $x$ missing values along the $L$-dimension), which frequently arise in practical applications. Moreover, it may also fail to comprehensively exploit the information from a complete data sample, because the corresponding conditional/marginal samples (already given within that joint sample) are not explicitly utilized.

Intuitively, one would prefer to use the available complete/incomplete data as they are (e.g., to avoid unintentional interventions that violate the i.i.d. assumption) and to comprehensively exploit the data information therein (e.g., to from multiple training tasks that encourage learning compositional intrinsic data knowledge in model parameters [37, 1]). The latter has been extensively demonstrated by existing foundation models, which exploit data information from many conditional perspectives in a multitasking manner and demonstrate impressive robustness and generalization. For clarity, Table I summarizes the commonly-used training objectives for foundation models, i.e., masked language modeling (LM), causal/auto-regressive LM, and permutation LM.

To comprehensively exploit the data information within both complete and incomplete samples, we propose the following unsupervised big learning that leverages a universal foundation model to model many/all joint/conditional/marginal data distributions\(^4\) simultaneously (manifested as “big” learning with massive data tasks).

\(^4\)A misunderstanding is that $p_\theta(y|x)$ has to be uni-model under the classification setup with feature $x$ and label $y$. Note a multi-mode model can have a uni-model practical instantiation.

\(^5\)The incomplete data are readily utilized in the corresponding conditional/marginal tasks.
where $q$ which is in the adversarial-learning territory with continuous observations, while the other is in the maximum-likelihood-learning territory with discrete observations.

Theorem 3.1 (Unsupervised/Uni-modal big learning). With the unsupervised/uni-modal setup, where a data sample $X = \{x\}$ contains only a feature $x \in \mathbb{R}^{L \times D}$ with length $L$, dimension $D$, the length index set $L = \{1, \ldots, L\}$, and any two non-overlapping subsets $S \subset \mathbb{L}$ and $T \subset \mathbb{L}$, $T \neq \emptyset$, unsupervised big learning leverages a universal foundation model $p_\theta(x_T|\mathbf{x}_S)$ (see Fig. 1a) to model many/all joint/conditional/marginal data distribution simultaneously, i.e.,

$$p_\theta(x_T|\mathbf{x}_S) \rightarrow q(x_T|\mathbf{x}_S), \forall (S, T) \in \Omega,$$

(3)

where $S$ and $T$ are named source/target index subsets, respectively, and $\Omega$ can be flexibly specified as the set that contains all $(S, T)$ pairs or a portion of them. For any $(S, T)$, $q(x_T|\mathbf{x}_S)$ denotes the corresponding joint/conditional/marginal data distribution, whose samples are readily selected from the whole available data.

Remark 3.2. In Theorem 3.1, $S \cup T$ need not be $\mathbb{L}$, meaning that incomplete data are naturally utilized.

Remark 3.3. Because source $\mathbf{x}_S$ and target $x_T$ may have different sizes for different $(S, T) \in \Omega$, one may prefer constructing $p_\theta(x_T|\mathbf{x}_S)$ of (3) as a Transformer/ViT to embrace its modeling flexibility.

Remark 3.4. The objective that measures the distance/divergence between both sides of (3) is application-dependent; possible choices include the cross-entropy loss, the GAN loss, energy-based models [33], diffusion models, score matching [24], etc.

Remark 3.5. Considering practical situations, one may alternatively or additionally do big learning in transformed domains, e.g., via $p_\theta(\hat{x}_T|\hat{x}_S)$ with $\hat{x} = g(x)$ or $p_\theta(h(x_T)|k(x_S))$ [21, 53], where $g(\cdot)$, $h(\cdot)$, and $k(\cdot)$ are domain-knowledge-inspired transformations.

3.2 Implementations of Unsupervised/Uni-modal Big Learning

We demonstrate unsupervised/uni-modal big learning with two example implementations, one of which is in the adversarial-learning territory with continuous observations, while the other is in the maximum-likelihood-learning territory with discrete observations.

3.2.1 Adversarial Learning for Foundation Models

Given continuous observations $x \in \mathbb{R}^{L \times D}$ (e.g., $x$ denoting a sequence of flattened patches of an image), we design a ViT-based universal model $p_\theta(x_T|\mathbf{x}_S)$ that models the generative processes of $x_T$ given $\mathbf{x}_S$ for all $(S, T)$ pairs (see Appendix C for the detailed architecture). The standard GAN loss [16] is employed as the objective that measures the divergence between both sides of (3).

Following (3) of Theorem 3.1, we propose to adversarially train the foundation model $p_\theta(x_T|\mathbf{x}_S)$ via minimizing the JS divergence between $p_\theta(x_T|\mathbf{x}_S)$ and $q(x_T|\mathbf{x}_S)$ for many/all $(S, T)$ pairs, i.e.,

$$\min \max_{\theta} \mathbb{E}_{q(S, T)}[\mathbb{E}_{q(x_S, x_T)} \log \sigma[f_\phi(x; S, T)] + \mathbb{E}_{p_\theta(x_T|\mathbf{x}_S)} q(x_S) \log \sigma[-f_\phi(x; S, T)],$$

(4)

where $q(S, T)$ denotes the sampling process of $(S, T)$ (see Appendix E for an example implementation), the discriminator $\sigma[f_\phi(x; S, T)]$ is also constructed as a ViT, and the optimal $f_\phi^*(x; S, T) = \frac{q(x_S, x_T)}{p_\theta(x_T|\mathbf{x}_S)} = \frac{q(x_S, x_T)}{p_\theta(x_T|\mathbf{x}_S)}$.

After training, the universal model $p_\theta(x_T|\mathbf{x}_S)$ is expected to possess many/all joint/conditional/marginal data capabilities. Note to naively collect all the capabilities, one need construct at least $N_{\text{all}} = \sum_{i=0}^{L-1} C_L^i \cdot (\sum_{k=1}^{L-1} C_L^k)$ models, which is clearly prohibitive. See Appendix A for details.
Table 1: Big learning and its special cases. In general, \( X = (y, x) \), \( x \in \mathbb{R}^{L \times D} \), \( y \in \mathbb{R}^{L' \times D'} \), \( L' = [L_p, L] \), \( S' = [S_p, S] \), and \( T' = [T_p', T] \); with \( X = (x) \) and \( L' = S' = T' = \emptyset \), unsupervised big learning is recovered. When \( y \in \{1, \ldots, C\}^{1 \times 1} \), it may represent a label. We ignore the implementation details and just focus on the core idea for demonstration.

| Big Learning | \( p_S(X_T|X_S) \rightarrow q_S(X_T|X_S) \), \( S' \in L' \), \( T' \in L' \) | \( S' \in L' \), \( T' \subseteq L' \) | \( S' \in L' \), \( T' \in L' \) | \( S' \in L' \), \( T' \in \emptyset \) | \( S' \in \emptyset \) |
|--------------|-------------------|-----------------|-----------------|-----------------|-----------------|
| Masked LM \( \text{ML} \) | \( E_{q(\mathcal{S}, \mathcal{T})} KL(q(x_S|x_T)||p(x_S|x_T)) \) | \( q(\mathcal{S}, \mathcal{T}) = \mathcal{U}(\mathcal{S}, \mathcal{T}) \), \( \mathcal{S} \) is a 85% random subset of \( L \), \( \mathcal{T} = L \setminus \mathcal{S} \) | \( p(x_S|x_T) = \Pi_{x \in \mathcal{S}} \text{Categorical}(x_S|p(x_S)) \) | \( \emptyset \) | \( \emptyset \) |
| Causal/Auto-regressive LM \( \text{CAR} \) | \( \sum_{x \subseteq \mathcal{S}, x \subseteq \mathcal{T}} E_{q(\mathcal{S}, \mathcal{T})} KL(q(x_S|x_T)||p(x_S|x_T)) \) | \( \mathcal{S} = \{(S_1, 1), (S_2, 2), \ldots\} \) \( \text{with } x \subseteq \mathcal{S} \) and \( x \subseteq \mathcal{T} \), \( \emptyset \) | \( \emptyset \) | \( \emptyset \) | \( \emptyset \) |
| Permutation LM \( \text{PL} \) | \( E_{q(\mathcal{S}, \mathcal{T})} \sum_{x \subseteq \mathcal{S}, x \subseteq \mathcal{T}} KL(q(x_S|x_T)||p(x_S|x_T)) \) | \( q(\mathcal{S}, \mathcal{T}) = \mathcal{U}(\mathcal{S}, \mathcal{T}) \), \( \mathcal{S} \) is a 85% random subset of \( L \), \( \mathcal{T} = 1 \) random permutation of \( L \setminus \mathcal{S} \) | \( \emptyset \) | \( \emptyset \) | \( \emptyset \) |
| MAE \( \text{MAE} \) | \( E_{q(\mathcal{S}, \mathcal{T})} KL(q(h(x_S)|x_T)||p(h(x_S)|x_T)) \) | \( q(\mathcal{S}, \mathcal{T}) = \mathcal{U}(\mathcal{S}, \mathcal{T}) \), \( \mathcal{S} \) is a 25% random subset of \( L \), \( \mathcal{T} = 1 \) random permutation of \( L \setminus \mathcal{S} \) | \( \emptyset \) | \( \emptyset \) | \( \emptyset \) |
| Big Learning with \( \text{CAR} \) \( \text{GL} \) | \( E_{q(\mathcal{S}, \mathcal{T})} KL(q(x_S|x_T)||p(x_S|x_T)) \) | \( q(\mathcal{S}, \mathcal{T}) = \mathcal{U}(\mathcal{S}, \mathcal{T}) \), \( \mathcal{S} \) is a random subset of \( L \), \( \mathcal{T} = 1 \) random permutation of \( L \setminus \mathcal{S} \) | \( \emptyset \) | \( \emptyset \) | \( \emptyset \) |
| Supervised Classification \( \text{SC} \) | \( E_{q(\mathcal{S}, \mathcal{T})} KL(q(x_S) || p(x_S)) \) | \( \emptyset = [0, 1], \emptyset = [L', 0], q(y|x) \) is e.g., a classifier | \( \emptyset \) | \( \emptyset \) | \( \emptyset \) |
| Joint Generation \( \text{JG} \) | \( E_{q(\mathcal{S}, \mathcal{T})} KL(q(y|x) || p(y|x)) \) | \( \emptyset = [0, \emptyset], \emptyset = [L', 0], p(x) \) may be a generator | \( \emptyset \) | \( \emptyset \) | \( \emptyset \) |
| Conditioned Generation \( \text{CG} \) | \( E_{q(\mathcal{S}, \mathcal{T})} KL(q(y|x) || p(y|x)) \) | \( \emptyset = [L', 0], \emptyset = [0, L], p(x|y) \) is a conditional flow | \( \emptyset \) | \( \emptyset \) | \( \emptyset \) |

In addition to \( 4 \), we further introduce training tasks by considering that any two model distributions \( p_S(x_T|x_S) \) and \( p_S(x_S|x_T) \) with \( S' \cup T' = S' \cup T' \) should be close to each other. Accordingly, we enable “communications” among any two functionalities of the one universal model \( p_S(x_T|x_S) \) via

\[
\min_{\Theta} \max_{\Phi} \mathbb{E}_{q(S', T') \in S'} \left[ \mathbb{E}_{p_S(x_{T_1}|x_{S_1})} \log \sigma[f_{\Phi}(x_{S_1}, S', T') - f_{\Theta}(x_{S_1}, S', T')] + \mathbb{E}_{p_S(x_{T_2}|x_{S_2})} \log \sigma[f_{\Phi}(x_{S_2}, S', T') - f_{\Theta}(x_{S_2}, S', T')] \right], \tag{5}
\]

where the “communication” discriminator can be implicitly constructed with the same neural network \( f_{\Phi}(x; S, T) \) from \( 4 \). Proofs are given in Appendix B.

Combining \( 4 \) and \( 5 \) yields an adversarial-learning implementation of the unsupervised big learning, which is the first principled adversarial pretraining strategy for foundation models, to our knowledge.

3.2.2 Maximum-Likelihood Implementation

Consider applications with discrete observations \( x \in \mathbb{Z}^{L \times 1} \), where, e.g., \( x \) denotes a sequence of text words or vector-quantized image patches \( [33] \). Equation \( 3 \) of Theorem \( 3 \) motivates us to model the distribution \( p_S(x_T|x_S) \) of multiple target words \( x_T \) conditioned on source words \( x_S \), which is challenging considering the correlations among \( x_T \)-words.

One solution to that challenge to leverage the causal LM to auto-regressively construct the model \( p_S(x_T|x_S) \) conditioned on \( x_S \), considering only the forward prediction ordering within \( x_T \). The core idea of unsupervised big learning, i.e., to exhaustively exploit data information, motivate us to step further and learn from various prediction orderings within \( x_T \) which is actually quite similar to the permutation LM \( 5 \) (see Table I).

With a Transformer-based universal model \( p_S(x_T|x_S) \) modeling the generative process of a target word \( x_T \) given source words \( x_S \) for any \( (S, T) \) pair, the unsupervised/uni-modal big learning yields

\[
\max_{\Theta} \mathbb{E}_{q(S, T) \in \Xi_{S, T}} \left[ \mathbb{E}_{q(x_S|x_T)} \log p_S(x_T|x_S) \right], \tag{6}
\]

where \( q(\mathcal{S}, \mathcal{T}) \) denotes the sampling process of \( (\mathcal{S}, \mathcal{T}) \) with random permutations, \( \mathcal{T} = \{t_1, t_2, \ldots\} \), \( \Xi_{S, T} = \{(S_1, t_1), (S_1, t_2), (S_1, t_3), (S_1, t_4), \ldots\} \), often \( p_S(x_T|x_S) = \text{Categorical}(x_T|p_S(x_S)) \) is modeled as a categorical distribution with probabilities \( p_S(x_S) \), and \( x_T \) always contain one word.

\( \text{The GAN implementation with } 4 \text{ and } 5 \text{ need not consider the ordering of } T \text{ thanks to its (conditionally) joint modeling nature.} \)
After unsupervised big learning, \( p_\theta(x_T^i|x_S^i) \) naturally brings versatile generation and data completion capabilities w.r.t. any predicting order.

### 3.3 Big Learning in General Settings

Thanks to the modeling flexibility of the unsupervised/uni-modal big learning with \( x \in \mathbb{R}^{L \times D} \), to generalize it into the big learning in general settings, where \( X = (y, x) \) contains an additional supervision \( y \in \mathbb{R}^{L' \times D''} \), is straight-forward. The key idea is to interpret paired multi-modal data as a “latter” sample.

**Theorem 3.6** (Big learning). With the general/multi-modal setup, where a data sample \( X = (y, x) \) contains both feature \( x \in \mathbb{R}^{L \times D} \) and its paired supervision \( y \in \mathbb{R}^{L' \times D''} \) with the \( X \)-length index set \( L' = [L', L] \), its any two non-overlapping source/target index subsets \( S' = [S', S] \) and \( T' = [T', T] \) with \( S' \subset L', \ T' \subset L', \text{ and } T' \neq 0 \), big learning leverages a universal foundation model \( p_\theta(X_T|X_S') \) (see Fig. 1B) to model many/all joint/conditional/marginal \( X \)-data distributions simultaneously, i.e.,

\[
p_\theta(X_T|X_S') \to q(X_T|X_S'), \forall (S', T') \in \Omega'
\]  

where \( \Omega' \) can be flexibly specified as the set that contains all \( (S', T') \) pairs or a portion of them. \( q(X) \triangleq q(y, x) \) is the underlying complete data distribution. For any \( (S', T') \), \( q(X_T|X_S') \) is the corresponding joint/conditional/marginal \( X \)-data distribution, whose samples are readily selected from the whole available data. Similar to Theorem 3.7 the objective that measures the distance/divergence between both sides of (7) is application-dependent.

**Remark 3.7.** For situations where \( X = (y, x) \) has the same data type (e.g., both \( y \) and \( x \) denote a sequence of continuous image-patches), big learning works basically the same as its unsupervised/uni-modal simplification in Theorem 3.6. However, for challenging situations where each modality has a different data type, e.g., where \( y \) denotes discrete text words but \( x \) are continuous image-patches, one may resort to the following two techniques to enjoy the big learning.

1. **To transform one data type into the other type for alignment**, e.g., one can vector-quantize the continuous \( x \) into a sequence of discrete tokens.

2. **To recursively reuse \( p_\theta(X_T|X_S') \) to isolate each type**, i.e., one can unfold the learning via

\[
p_\theta(X_T|X_S') = p_\theta(y_T|x_T, X_S')p_\theta(x_T|X_S') = p_\theta(X_T^i|X_{T, S'}^i)p_\theta(X_T|X_{S'}),
\]

where \( X_T^i/X_{S'}^i \) has one unique data type after unfolding. One can then resort to big learning both \( p_\theta(X_{T^i}|X_{T, S'}^i) \to q(X_T^i|X_{T, S'}^i) \) and \( p_\theta(X_T|X_{S'}) \to q(X_T|X_{S'}) \) for training.

**Remark 3.8.** From the learning perspective, big learning unifies (and contains as special cases) many machine learning paradigms within one framework, as detailed in Table 1. That generalizability of big learning, combined with its data/task flexibilities, may enable flexible combinations and communications among different learning paradigms.

### 3.4 Discussions on Big Learning

Without loss of generality, we focus on the simplified notations in the unsupervised settings for presentation and only employ the complicated ones in Theorem 3.6 if necessary.

**Can we share one universal foundation model \( p_\theta(x_T|x_S) \) among all \( (S, T) \) pairs?** Yes, and it’s what we should do. All conditional/marginal distributions \( q(x_T|x_S) \) can be analytically derived from the joint one \( q(x) \), meaning that they all share the same set of underlying “parameters”\(^8\); accordingly, the corresponding joint/conditional/marginal modelings are also expected to share parameters. Besides, sharing parameters also enables self-regularization among joint/conditional/marginal modelings, which likely encourages model parameters to approach that underlying “parameters.”

**On big-learned model parameters and latent features.** As aforementioned, most exiting foundation models that exhibit extraordinary robustness, adaptability, and generalization are special cases of the big learning. Accordingly, we try to explain from the big learning perspective why they have such amazing characteristics.

\(^8\text{We present with two modalities for simplicity; the presented big learning can be readily generalized to situations with multiple paired modalities.}\)
• Firstly, by referring to (3) and (7), both the model parameters and its latent features are shared among many/all joint/conditional/marginal data modeling tasks, which have the same consistent goal of modeling the intrinsic data information (i.e., the aforementioned underlying “parameters” of $q(x_T|\mathbf{x}_S)$) from diverse perspectives. Therefore, it’s expected that big learning would encourage summarizing intrinsic compositional data knowledge \cite{55, 37} in the model parameters (and its latent features), manifested as those amazing characteristics.

• Secondly, the extraordinary data and task flexibilities of big learning enable large-scale training with massive complete/incomplete data and diverse tasks (across potentially many domains). The significantly expanded training experiences (associated with both data and tasks) are expected to effectively reduce the training-test (or pretraining-finetuning) gap and therefore improve the robustness/generalization of big-learned foundation models.

Big learning versus self-supervised contrastive learning. Contrastive learning focuses on exploiting domain prior knowledge to learn generally applicable data representations for downstream tasks \cite{22, 9, 17, 10}. From the perspective of prior exploitation, contrastive learning is orthogonal to the big learning that is mostly data-driven. One can of course consider leveraging the flexibility of big learning to combine it with contrastive learning to incorporate trustworthy domain priors.

4 Experiments

The data/task flexibilities of the big learning significantly expand its scope of application, which, however, also brings tremendous challenges to comprehensive evaluations of its full potential.

Due to our limited computation budget, we concentrate on demonstrating the characteristics of the big learning from diverse perspectives. Specifically, we first reveal that unsupervised big learning is indeed capable of delivering all joint/conditional/marginal data capabilities, leveraging the adversarial-learning implementation with (4) and (5) on the MNIST/CelebA datasets (see Appendix E for details). We then demonstrate the generalization capability of that big-learned foundation model with diverse abused out-of-domain challenges. Next, based on the maximum-likelihood implementation in (6), we show that big learning can naturally handle multi-modal data and its joint/conditional/marginal data capabilities directly manifest as versatile functions of great practical value, e.g., classification and generation. Finally, considering the quantitative evaluations of the big learning, we conduct experiments on the GLUE benchmark to reveal that big learning can serve as a superior fine-tuning strategy than the naive one.

It’s worth noting that current big/foundation models—the majority of which are the special cases of the big learning—have offered copious evidence for the effectiveness of the big learning, from a variety of special-case perspectives. Moreover, the co-occurrent work \cite{3} has provided concrete empirical proofs that fitting all joint/conditional/marginal distributions with one (diffusion) model (i.e., performing the big learning) could deliver SOTA performance in representative tasks like text-to-image generation.

4.1 Versatile Data Completion Capabilities With Adaptive Generation Diversity

We first test the big-learned data generation/completion capabilities with different ratios $r_S$ of $\mathbf{S}$ in $\mathbf{L}$. For a specific $r_S$, we either randomly sample $r_S L$ image patches or choose the first $r_S$-portion to form the source $\mathbf{x}_S$, which is then input to the model $p_\theta(x_T|\mathbf{x}_S)$ for image completion.
4.2 Generalization on Abused Anomalous Out-Of-Domain Completion

We design abused completion tasks to test the generalization of the big learning. Specifically, we intentionally design $x_S$ with (i) abused interventions to source patches (e.g., random relocation and duplication, as shown in Fig. 4(a)); (ii) mixed-up patches from different data samples (see Fig. 4(b)); and (iii) unseen out-of-domain image patches, as shown in Fig. 4(c).

It’s clear that big learning manages to handle these abused $x_S$ with reasonable image completion; e.g., see the realistic characters with overall consistent style and smooth strokes in Fig. 4(a), the harmoniously completed faces even with mismatched face frame and hair color in Fig. 4(b), and the relatively smooth out-of-domain completion in Fig. 4(c). These surprising results from abused anomalous out-of-domain completions (along with the great successes of existing foundation models) validate the generalization capability of the presented big learning.
4.3 Leveraging Big Learning to Unify Classification and Generation

We test the big learning in the general settings, where $X = (y, x)$ contains both image tokens $x \in \mathbb{Z}^{L \times 1}$ and a paired label $y \in \{1, \cdots, C\}^{1 \times 1}$. We conduct the experiment on MNIST and follow [4, 43] to first vector-quantize an image for its tokens $x$, followed by big learning based on (6). Details are given in Appendix F.

Given the big-learned universal model $p_\theta(X_T | X_S')$, one can retrieve from it versatile data capabilities by specifying the corresponding $(S', T')$, such as joint generation (i.e., $p_\theta(x)$; see Fig. 5(a) for the results), label-conditioned generation (i.e., $p_\theta(x|y)$; see Fig. 5(b)), classification (i.e., $p_\theta(y|x)$), random completion (i.e., $p_\theta(x_T|x_S)$), label-conditioned completion (i.e., $p_\theta(x_T|x_S, y)$), etc. These simultaneously-delivered capabilities are likely valuable for counterfactual analysis and reasoning.

4.4 Quantitative Evaluations on the GLUE Benchmark

Because of our limited computation budget, we cannot afford to make systematic quantitative comparisons between the big learning and existing methods on pretraining a foundation model with large-scale data. Alternatively, we empirically reveal that the big learning is a superior fine-tuning strategy than the naive one.

Specifically, we initialize with the pretrained xlnet-base-cased model from the Hugging Face transformers library [54] and then test fine-tuning it on downstream RTE/MRPC/SST-2 tasks (from the GLUE Benchmark [51]) with (i) the naive fine-tuning strategy (termed FT) and (ii) the big learning (termed big-learn), respectively. Table 2 summarizes the quantitative evaluation results, where it’s clear that big-learn consistently outperforms FT, even without careful tuning. See Appendix G for details.

5 Conclusions

We propose the big learning that exhaustively exploits the available data information and potentially delivers all joint/conditional/marginal data capabilities. We reveal that big learning (i) comes with extraordinary training flexibilities for complete/incomplete data and for customizing training tasks, (ii) contains most existing foundation models as special cases, and (iii) unifies conventional machine learning paradigms and enables their flexible cooperations. Though inspiring, big learning also shares the same constraints of foundation models [5, 57], e.g., either to comprehensively reveal its potential or to verify its effectiveness (for general downstream tasks) is extremely challenging and time-consuming. Therefore, we believe that big learning needs our community and vice versa.
References

[1] Armen Aghajanyan, Anchet Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning. arXiv preprint arXiv:2101.11038, 2021.

[2] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. arXiv preprint arXiv:2202.03555, 2022.

[3] Fan Bao, Shen Nie, Kaiven Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. arXiv preprint arXiv:2303.06555, 2023.

[4] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254, 2021.

[5] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258, 2021.

[6] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In ICLR, 2019.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. NeurIPS, 33:1877–1901, 2020.

[8] Mark Chen, Jerry Tvorek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374, 2021.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In ICML, pages 1597–1607. PMLR, 2020.

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In CVPR, pages 15750–15758, 2021.

[11] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In ICCV, pages 9640–9649, 2021.

[12] B. Dai and D. Wipf. Diagnosing and enhancing VAE models. In ICLR, 2019.

[13] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[14] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516, 2014.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth a 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NeurIPS, pages 2672–2680, 2014.

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Ghehslaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. NeurIPS, 33:21271–21284, 2020.

[18] Tanmay Gupta, Amita Kamath, Aniruddha Kembhavi, and Derek Hoiem. Towards general purpose vision systems. arXiv preprint arXiv:2104.00743, 2021.

[19] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In CVPR, volume 2, pages 1735–1742. IEEE, 2006.

[20] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. arXiv preprint arXiv:2104.05704, 2021.

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377, 2021.

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In CVPR, pages 9729–9738, 2020.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. NeurIPS, 33:6840–6851, 2020.

[24] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. Journal of Machine Learning Research, 6(4), 2005.
[25] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. NeurIPS, 34, 2021.

[26] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In CVPR, June 2019.

[27] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In CVPR, pages 8110–8119, 2020.

[28] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. NeurIPS, 33:12104–12114, 2020.

[29] Tero Karras, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. NeurIPS, 34, 2021.

[30] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[31] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. NeurIPS, 31, 2018.

[32] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942, 2019.

[33] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. 2022.

[34] Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhaowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. arXiv preprint arXiv:2107.04589, 2021.

[35] Qing Li, Boqing Gong, Yin Cui, Dan Kondratyuk, Xianzhi Du, Ming-Hsuan Yang, and Matthew Brown. Towards a unified foundation model: Jointly pre-training transformers on unpaired images and text. arXiv preprint arXiv:2112.07074, 2021.

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

[37] Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Pretrained transformers as universal computation engines. arXiv preprint arXiv:2103.05247, 2021.

[38] L. Mescheder, A. Geiger, and S. Nowozin. Which training methods for GANs do actually converge? In ICML, pages 3478–3487, 2018.

[39] OpenAI. Chatgpt: Optimizing language models for dialogue, 2022.

[40] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155, 2022.

[41] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. OpenAI Blog, 1(8):9, 2019.

[42] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125, 2022.

[43] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In ICML, pages 8821–8831. PMLR, 2021.

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10684–10695, 2022.

[45] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. arXiv preprint arXiv:2205.11487, 2022.

[46] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modelling through stochastic differential equations. arXiv preprint arXiv:2011.13456, 2020.

[47] A. Stickland and I. Murray. BERT and PALS: Projected attention layers for efficient adaptation in multi-task learning. arXiv preprint arXiv:1902.02671, 2019.

[48] Alex Tamkin, Vincent Liu, Rongfei Lu, Daniel Fein, Colin Schultz, and Noah Goodman. DABS: A domain-agnostic benchmark for self-supervised learning. arXiv preprint arXiv:2111.12062, 2021.

[49] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In ICCV, pages 32–42, 2021.
[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In NeurIPS, pages 5998–6008, 2017.

[51] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In ICLR, 2018.

[52] Wen Wang, Jing Zhang, Yang Cao, Yongliang Shen, and Dacheng Tao. Towards data-efficient detection transformers. In European conference on computer vision, pages 88–105. Springer, 2022.

[53] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. arXiv preprint arXiv:2112.09133, 2021.

[54] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[55] Yuhuai Wu, Markus N Rabe, Wenda Li, Jimmy Ba, Roger B Grosse, and Christian Szegedy. Lime: Learning inductive bias for primitives of mathematical reasoning. In ICML, pages 11251–11262. PMLR, 2021.

[56] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. Le. XLNet: Generalized autoregressive pretraining for language understanding. In NeurIPS, pages 5753–5763, 2019.

[57] Sha Yuan, Hanyu Zhao, Shuai Zhao, Jiahong Leng, Yangxiao Liang, Xiaozhi Wang, Jifan Yu, Xin Lv, Zhou Shao, Jiaao He, et al. A roadmap for big model. arXiv preprint arXiv:2203.14101, 2022.

[58] Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. StyleSwin: Transformer-based gan for high-resolution image generation. arXiv preprint arXiv:2112.10762, 2021.

[59] Long Zhao, Zizhao Zhang, Ting Chen, Dimitris Metaxas, and Han Zhang. Improved transformer for high-resolution gans. NeurIPS, 34, 2021.
Appendix of Big Learning

A  On naive modeling of all joint/conditional/marginal data distributions

We present with the unsupervised settings, where \( x \in \mathbb{R}^{L \times D} \) with length \( L \) and dimension \( D \) (like \( L \) flattened patches of an image or \( L \) words with \( D = 1 \)). It’s straightforward to generalize the following analyses to the general settings with a data sample \( X = (y, x) \) contains an additional supervision \( y \in \mathbb{R}^{L_y \times D_y} \). Considering \( D > 1 \) and \( D = 1 \) for image patches and text words, respectively, we concentrate on analyzing the modeling of all joint/conditional/marginal data distributions \( w.r.t. \) the length \( L \) below.

As mentioned in the main manuscript, one need construct \( N_{\text{all}} = \sum_{i=0}^{L-1} C_L^i (\sum_{k=1}^{L-i} C_L^k) \) models to naively model all joint/conditional/marginal data distributions, to collect all joint/conditional/marginal data capabilities. \( C_L^i \) denotes the number of \( i \)-combinations from a set with \( L \) elements.

To elaborate on that, consider a simple \( k \)-ordered arrangements of \( S \). In that case, to naively modeling all joint/conditional/marginal data distributions, \( \forall \) \( y \in \mathbb{R} \), and the length index set \( L = \{1, 2, 3\} \).

- The goal of the joint modeling is to deliver \( p_\theta(x) \rightarrow q(x) \) with one model \( p_\theta(x) \).

- By contrast, to naively model all joint/conditional/marginal data distributions, one need construct \( 19 \) models for such a simple \( 3 \)-length problem, \( i.e., \)

\[
\begin{align*}
& p_\theta(x_1), \ p_\theta(x_2), \ p_\theta(x_3), \ p_\theta(x_1, x_2), \ p_\theta(x_2, x_3), \ p_\theta(x_1, x_3), \ p_\theta(x_1, x_2, x_3), \\
& p_\theta(x_2|x_1), \ p_\theta(x_3|x_1), \ p_\theta(x_2|x_3), \ p_\theta(x_1, x_2|x_1), \ p_\theta(x_3|x_2), \\
& p_\theta(x_1|x_2, x_3), \ p_\theta(x_1|x_3), \ p_\theta(x_2|x_3), \ p_\theta(x_1, x_2|x_3).
\end{align*}
\]

(9)

Based on the above \( 3 \)-length problem, one can readily summarize the following two steps in calculating the number of models in naively modeling all joint/conditional/marginal data distributions, \( i.e., \)

\( q(x_T|x_S), \forall S \subseteq L, T \subseteq L, T \neq \emptyset \).

1. **Sample \( S \)**. The source index set \( S \) may contain \( \{0, \ldots, L - 1\} \) indexes/locations, where \( S \) containing \( 0 \) index corresponds to joint/marginal generations and \( S \) containing \( \geq 1 \) indexes corresponds to conditional generations/completions. For a special case with \( i \) indexes in \( S \) with \( i \in [0, L - 1] \), one has \( C_L^i \) ways to specify that source index set \( S \).

2. **Sample \( T \) conditioned on \( S \)**. Given a \( S \) consisting of \( i \) indexes, the target index set \( T \) could contain \( \{1, \ldots, L - i\} \) indexes/locations outside \( S \). For a special case of \( T \) containing \( k \) indexes where \( k \in \{1, L - i\} \), one has \( C_{L-k}^k \) ways to specify the target \( T \).

Therefore, to naively model all joint/conditional/marginal data distributions, one need construct \( N_{\text{all}} = \sum_{i=0}^{L-1} C_L^i (\sum_{k=1}^{L-i} C_L^k) \) models, which, however, is prohibitive in practice.

Note with ideal modeling of \( q(x_T|x_S) \), the orders in \( S/T \) should not matter. However, that may not hold true considering practical constraints, \( e.g., \) where existing joint modeling techniques fail to model the multi-mode characteristics of \( x_T \). Besides, in the NLP application of language modeling, one may be interested in versatile (conditional) generation ordering (as defined in \( T \)), mimicking the permutation language modeling \( [56] \). In that case, to naively modeling all joint/conditional/marginal data distributions, one need construct \( N_{\text{all}}' = \sum_{i=0}^{L-1} C_L^i (\sum_{k=1}^{L-i} A_{L-k}^k) \) models to take into consideration the order of \( T \), where the order of \( S \) is ignored and \( A_{L-k}^k \) denotes the number of the ordered arrangements of \( k \) elements from a set with \( L - 1 \) elements. Similarly, one need construct \( N_{\text{all}}'' = \sum_{i=0}^{L-1} A_L^i (\sum_{k=1}^{L-i} A_{L-k}^k) \) models to model the orders in both \( S \) and \( T \).
At first sight of Eqs. (12) and (13), it seems one should at least construct two discriminators, with

\[ f(x) \]

To leverage the GAN training framework \cite{16}, one needs the sampling capabilities from the distributions of interest. With \( T = L \setminus S \), here we are interested in the joint distributions with accessible sampling capabilities, including

\[ q(x) \]

Note one can of course exploit the flexibility of big learning to define other joint distributions with sampling capabilities, including

\[ x \]

Here we present the detailed derivations/proofs for the GAN example associated with Eqs. (4) and (5).

B Derivations of the GAN example associated with Eqs. (4) and (5)

Here we present the detailed derivations/proofs for the GAN example associated with Eqs. (4) and (5) of the main manuscript. For better understanding, we begin with a simplified case where \( T = L \setminus S \), followed by generalizing the results to the general situations with \( T \subseteq L \setminus S \).

B.1 \( T = L \setminus S \)

To leverage the GAN training framework \cite{16}, one needs the sampling capabilities from the distributions of interest. With \( T = L \setminus S \), here we are interested in the joint distributions with accessible sampling capabilities, including

\[ q(x) \]

Note one can of course exploit the flexibility of big learning to define other joint distributions with sampling capabilities, such as an recursively defined distribution

\[ p_\theta(x; S^1, S^2) = p_\theta(x_{L \setminus S^2} | x_{S^2}) p_\theta(x_{S^2}), \]

where \( p_\theta(x_{S^2}) = \int p_\theta(x_{L \setminus S^1} | x_{S^1}) q(x_{S^1}) dx_{L \setminus S^1} \). For simplicity, we focus on the simplified settings in Eq. (10) and leave the interesting but complicated recursive case for future research.

Given the underlying data distribution \( q(x) \) and “model” distributions \( p_\theta(x; S) \) in Eq. (10),

1. one can match any \( p_\theta(x; S) \) to \( q(x) \) adversarially with a GAN. Take the standard GAN \cite{16} for an example, the objective is

\[ \min_\theta \max_\phi \mathbb{E}_{q(x)} \log \sigma(f_\phi(x; S)) + \mathbb{E}_{p_\theta(x_{L \setminus S} | x_{S})} \log(1 - \sigma(f_\phi(x; S))), \]

where the optimal \( f_\phi^*(x; S) = \frac{\sigma(f_\phi(x; S))}{\mathbb{E}_{p_\theta(x_{L \setminus S} | x_{S})} \log(1 - \sigma(f_\phi(x; S)))} \). Ideally, optimizing the above objective is identical to minimizing the Jensen-Shannon divergence \( JS[q(x) || p_\theta(x; S)] \), as illustrated with the blue solid arrows in Fig. 6.

2. one can also conduct matching among any two model distributions (e.g., \( p_\theta(x; S^1) = p_\theta(x_{L \setminus S^1} | x_{S^1}) q(x_{S^1}) \) and \( p_\theta(x; S^2) = p_\theta(x_{L \setminus S^2} | x_{S^2}) q(x_{S^2}) \)) to enable communications/cooperations among them, via optimizing

\[ \min_\theta \max_\phi \left\{ \mathbb{E}_{p_\theta(x_{L \setminus S^1} | x_{S^1})} \log \sigma(f_\phi^*(x; S^1, S^2)) + \mathbb{E}_{p_\theta(x_{L \setminus S^2} | x_{S^2})} \log(1 - \sigma(f_\phi^*(x; S^1, S^2))) \right\} \]

where the optimal \( f_\phi^*(x; S^1, S^2) = \frac{\sigma(f_\phi(x; S^1, S^2))}{\mathbb{E}_{p_\theta(x_{L \setminus S^1} | x_{S^1})} \log(1 - \sigma(f_\phi(x; S^1, S^2)))} \). The orange dotted arrows in Fig. 5 demonstrate such idea.

At first sight of Eqs. (12) and (13), it seems one should at least construct two discriminators, with \( f_\phi(x; S) \) and \( f_\phi^*(x; S^1, S^2) \) respectively. However, we notice that

\[ f_\phi^*(x; S^1, S^2) = \log \frac{q(x)}{p_\theta(x_{L \setminus S^2} | x_{S^2}) q(x_{S^2})} - \log \frac{q(x)}{p_\theta(x_{L \setminus S^1} | x_{S^1}) q(x_{S^1})} = f_\phi^*(x; S^2) - f_\phi^*(x; S^1). \]
Accordingly, we propose to employ further simplification that builds $f'_\phi(x; S^1, S^2)$ on top of $f_\phi(x; S)$, i.e., we reformulate Eq. (13) as

$$
\min_{\theta} \max_{\phi} \left\{ \mathbb{E}_{p_{x\mid x_1} \mid x_2 \mid q(x_2)} \log \sigma [f_\phi(x; S^2) - f_\phi(x; S^1)] + \mathbb{E}_{p_{x\mid x_1} \mid x_2 \mid q(x_2)} \log \sigma [f_\phi(x; S^1) - f_\phi(x; S^2)] \right\}. 
$$

(14)

Till now, we present the derivations associated with $T = L \setminus S$, i.e., matching in the joint space. In what follows, we generalize to the settings with $T \subseteq L \setminus S$, to deliver (unsupervised) big learning in all joint/conditional/marginal spaces.

**B.2 $T \subseteq L \setminus S$**

Similar to the previous section, we also consider simplified situations with no recursiveness, that is, we do not consider a model distribution $p_\theta(x_T \mid x_S)p_\theta(x_S)$, even though such recursive flexibility of big learning is quite interesting. We leave that as future research.

Accordingly, the considered joint/conditional/marginal distributions with sampling capabilities are

$$
q(x_{S \cup T}) = p_\theta(x_T \mid x_S)q(x_S) \quad \forall S, T
$$

(15)

where $S \cup T$ need not be $L$. Note $S \cup T \subset L$ means the corresponding $q(x_{S \cup T})$ is a *marginal* data distribution, whose data samples are readily accessible from those of $q(x)$.

Similar to the previous section,

- one can match any model distribution $p_\theta(x_{S \cup T})$ to the underlying joint/marginal data distribution $q(x_{S \cup T})$, via the standard GAN objective

$$
\min_{\theta} \max_{\phi} \mathbb{E}_{q(x_{S \cup T})} \log \sigma [f_\phi(x; S, T)] + \mathbb{E}_{p_{x\mid x_1 \mid x_2 \mid q(x_2)}} \log (1 - \sigma (f_\phi(x; S, T))),
$$

(16)

where $f_\phi(x; S, T) = \log \frac{q(x_{S \cup T})}{p_\theta(x_T \mid x_S)q(x_S)} = \log \frac{q(x_T \mid x_S)}{p_\phi(x_T \mid x_S)}$.

- one can also conduct matching among any two model distributions, e.g., $p_\theta(x_{T_1} \mid x_S)q(x_{S_1})$ and $p_\theta(x_{T_2} \mid x_S)q(x_{S_2})$, as long as $S^1 \cup T^1 = S^2 \cup T^2$, with the corresponding objective

$$
\min_{\theta} \max_{\phi} \left\{ \mathbb{E}_{p_{x\mid x_1 \mid x_2 \mid q(x_2)}} \log \sigma (f'_\phi(x; S^1, T^1, S^2, T^2)) + \mathbb{E}_{p_{x\mid x_1 \mid x_2 \mid q(x_2)}} \log (1 - \sigma (f'_\phi(x; S^1, T^1, S^2, T^2))) \right\},
$$

(17)

where $f'_{\phi \cdot} (x; S^1, T^1, S^2, T^2) = \log \frac{p_\theta(x_{T_2} \mid x_S)q(x_{S_2})}{p_\theta(x_{T_1} \mid x_S)q(x_{S_1})} = f_{\phi \cdot} (x; S^2, T^2) - f_{\phi \cdot} (x; S^1, T^1)$.

For further simplifications, we again resort to

$$
f'_{\phi \cdot} (x; S^1, T^1, S^2, T^2) = \log \frac{q(x_{S^1 \cup T^2})}{p_\theta(x_{T_2} \mid x_S)q(x_{S_2})} - \log \frac{q(x_{S^1 \cup T^2})}{p_\theta(x_{T_1} \mid x_S)q(x_{S_1})} = f_{\phi \cdot} (x; S^2, T^2) - f_{\phi \cdot} (x; S^1, T^1)
$$

and build $f'_{\phi \cdot} (x; S^1, T^1, S^2, T^2)$ on top of $f_{\phi \cdot} (x; S, T)$.

Accordingly, Eq. (17) is reformulated as

$$
\min_{\theta} \max_{\phi} \left\{ \mathbb{E}_{p_{x\mid x_1 \mid x_2 \mid q(x_2)}} \log \sigma (f_{\phi \cdot} (x; S^2, T^2) - f_{\phi \cdot} (x; S^1, T^1)) + \mathbb{E}_{p_{x\mid x_1 \mid x_2 \mid q(x_2)}} \log \sigma (f_{\phi \cdot} (x; S^1, T^1) - f_{\phi \cdot} (x; S^1, T^1)) \right\},
$$

(18)

Accordingly, we conclude the proofs for the GAN example of the main manuscript.
Figure 7: Unsupervised big learning (a) and its special cases (b). Often a mask token [M] is inserted to the input locations outside S for forward propagation, while no loss is back-propagated to the output locations outside T. Note inserting the [M] tokens later in a middle layer (but at the same location) often lightens the computation and memory burdens but improves the performance [21].

Figure 8: Big learning (a) and its special case of BERT (b). Similar to the mask token [M] for \( x \) (see Fig. 7b), we employ another mask token \([M_y]\) for \( y \), which works identically to the classification token [CLS] in BERT settings [47] and the start-of-sentence token in GPT settings [7]. Often inserting \([M]/[M_y]\) tokens later in a middle layer improves performance [21,49].

C On model architectures of the GAN example in Eqs. (4) and (5)

We next focus on discussing the model architectures of the GAN generator and discriminator employed in Eqs. (16) and (18) (i.e., Eqs. (4) and (5) of the main manuscript).

Recently, the community begins to exploit integrating ViTs into GANs [25,34,59,58]. For example, the ViTGAN [34], delivering SOTA generative performance, employs simple modifications to the ViT architecture to construct the generator and the discriminator, but adopts many techniques to regularize the ViT-based discriminator for stable training. Motivated by the modeling flexibility of ViTs, we also employ ViT-based GAN generator and discriminator in the experiments, but similarly, find it challenging to stabilize GAN training with a ViT-based discriminator. It’s worth highlighting that it’s possible to design other alternative model architectures for the big learning; we employ what’s presented below for a demonstration.
Fig. 9 demonstrates the employed GAN generator and discriminator, both of which are constructed with Transformers/ViT architectures to exploit their modeling capabilities and flexibilities.

- **GAN Generator.** Following the MAE [21], we design the GAN generator \( p_\theta(x_T | x_S) \) with an autoencoder-like architecture, which employs an encoding G-Encoder and a decoding G-Decoder, as shown in Fig. 9a. The G-Encoder encodes the source patches \( x_S \) (if any) to their latent codes; then, these codes are combined with the mask tokens \([M]\), patch-wise noise embeddings, and new positional encodings to serve as the input of the G-Decoder; finally, the G-Decoder transforms its input to generate the target patches \( x_T \).

\([M]\) tokens are inserted later in a middle layer, because doing this often improves performance and lowers the computational burden [21]. A noise \( z \) is mapped with an 8-layer MLP to produce the patch-wise noise embeddings \( \{n_1, \cdots, n_L\} \). Note we also introduce another token \([M_n]\) to indicate no noise embeddings are necessary at the corresponding source locations in \( S \).

- **GAN Discriminator.** As shown in Fig. 9b, we also modify the Transformer/ViT architecture to construct the universal GAN discriminator \( \sigma(f_\phi(x; S, T)) \) that applies to all \((S, T)\) cases.
We employ an additional CLS token mimicking the BERT, whose output indicates whether the input patches are realistic or not (more specifically, whether they form a “real” data from \(q(x_{S \cup T})\) or a fake one from \(p_{\theta}(x_{T} \mid x_{S}) q(x_{S})\), by referring to Eq. (16)). The input of the discriminator consists of patch embeddings, positional embeddings, and two new special tokens ([M_s] and [M_t]) that indicate source or target patches mimicking the sentence tokens in the BERT.

D Big learning versus contrastive learning

Contrastive learning [19] aims at learning a latent representation space, where the representations of different views of the same image (“positive pairs”) are near each other but those from different images (“negative pairs”) are far away from each other.

As discussed in the main manuscript, the self-supervised contrastive learning focuses on exploiting domain prior knowledge to learn generally applicable data representations, while the presented big learning is mostly data-driven. From that perspective, they are orthogonal to each other. However, we reveal below that big learning and contrastive learning have a lot in common.

- Both of them are based on massive multi-task training, associated with source/target indexes \((S, T)\) and online/target augmentation pairs \((A, B)\) (see Fig. 10a), respectively.
- Both of them share a universal model among massive training tasks.
- From the information perspective, both of them predict (or retrieve) the \(T/B\)-associated information conditioned on the information related to \(S/A\), as detailed below.

(a) Group 1 predicts/retrieves the positively paired sample from the negative samples within a mini-batch. \(A\) and \(B\) denote the online and target augmentation, respectively.

(b) Group 2 directly generates/predicts the target/teacher projection conditioned on the student projection.

Figure 10: Demonstrations of contrastive learning methods. (b) is adapted from Fig. 3 of [10].
Existing contrastive learning methods can be roughly grouped into two groups, based on whether the method uses negative pairs (like SimCLR [9] and MoCo [11]) or not (like BYOL [17] and SimSiam [10]).

- **Group 1.** Contrastive learning methods using negative pairs, like SimCLR and MoCo, can be interpreted as retrieving (in the latent representation space) the target positively paired sample $B_i$ from the negative samples within the mini-batch, conditioned on the source augmented sample $A_i$, as illustrated in Fig. 10a.

- **Group 2.** Contrastive learning methods not using negative pairs, like BYOL and SimSiam, directly predict/generate (in the latent representation space) the target/teacher projection associated with $B_i$, conditioned on the student projection associated with $A_i$, as demonstrated in Fig. 10b.

Either group of contrastive learning methods retrieves or predicts the $B$-associated information conditioned on the information related to $A$, which is quite similar to the proposed big learning that predicts/generates $x^T$ conditioned on $x^S$. Therefore, from the information perspective, both of them predict (or retrieve) a piece of data/prior information conditioned on another piece of data/prior information.

It’s interesting to consider combining big learning with contrastive learning to exhaustively exploit the available information from both data and domain-prior perspectives.

### E Experimental settings used in Sections 4.1 and 4.2 of the main manuscript

We employ the same model architectures in the previous Section C for the experiments on the MNIST and CelebA datasets, with the detailed hyperparameters summarized in Table 3. Despite the relatively small models used, we find that big learning is capable of delivering potentially all joint/conditional/marginal data capabilities simultaneously. We adopt the AdamW optimizer [36] with $\beta = (0.1, 0.999)$ and constant learning rates for both the generator and the discriminator. Code will be released upon publication.

| Dataset   | MNIST | CelebA |
|-----------|-------|--------|
| Image size| 64    | 120    |
| Patch size| 8     | 10     |
| G-Encoder depth | 6      | 6      |
| G-Encoder #heads  | 8      | 8      |
| G-Encoder dim    | 256    | 256    |
| G-Decoder depth  | 6      | 6      |
| G-Decoder #heads  | 8      | 8      |
| G-Decoder dim    | 512    | 512    |
| D depth          | 6      | 6      |
| D #heads         | 8      | 8      |
| D dim            | 256    | 256    |
| $\lambda_{GP}$   | real   | real   |
| Learning rate    | $10^{-4}$ | $10^{-4}$ |
| Batch size       | 256    | 128    |
| Source ratio $\|z^S\|/\|L\|$ | Beta(0.5,3) | Beta(0.5,3) |
| Target ratio $\|z^S\|/\|L\|/\|z^S\|$ | Beta(3.0,5) | Beta(3.0,5) |
| Communication source ratio $\|z^S\|/\|L\|/\|z^S\|/\|z^T\|$ | Beta(0.5,3) | Beta(0.5,3) |

Overall, we find it’s quite straightforward to implement the MNIST experiments with the standard implementations discussed in Sections B and C, without resorting to any “tricks” like warm-up or gradient clipping. However, on the more complicated CelebA experiments, we find it’s necessary to employ some, as detailed below.
• We employ warm-up in the first 10 epochs for both the GAN generator and discriminator; after that, we use the constant learning rate given in Table 3.

• We apply gradient clipping, with the max norm of 5, to both the generator and discriminator optimizers.

• Similar to [34], we also find it challenging to stabilize GAN training with a ViT-based discriminator. To deal with that, we additionally (i) overlap image patches [34] with e.g., 2 pixels at the input of the discriminator (different from the non-overlapping image patches used in the vanilla ViT); and (ii) use a larger hyperparameter $\epsilon = 10^{-5}$ in the AdamW optimizer.

Other empirical experiences are listed below.

• We empirically find that the last normalization layers of both the GAN generator and discriminator have a significant influence on the learning stability and final performance. Specifically, replacing the last LayerNorm of the G-Decoder of the generator with a LeakyReLU leads to improved generative performance, whereas replacing the last LayerNorm of the discriminator with other normalization/activation layers results in training collapse.

• Employing an additional convolutional head (like a 3-layer CNN) to the output of the generator often leads to improved performance and training stability.

• Instead of only introducing noise embeddings at the first layer of the G-Decoder of the generator, as shown in Fig. 9a, we find it’s beneficial to concatenate the same set of noise embeddings layer-wisely into the G-Decoder layers.

F Big Learning Unifies Classification and Generation

After following [4, 43] to vector-quantize an image into discrete tokens $x \in \mathbb{Z}^{L \times 1}$, the observed random variable $X = (y, x)$ with discrete label $y$ now has only one data type. Accordingly, one can readily generalize Eq. (6) of the uni-model unsupervised big learning to solve the problem. Specifically, with a Transformer-based universal model $\theta(X_{T'}|X_{S'})$ that models the generative process of a target token $X_{T'}$ given source ones $X_{S'}$ for any $(S', T')$ pair, the big learning yields

$$\max_{\theta} \mathbb{E}_{q(S', T')} \sum_{(S', T') \in \Xi'_{S', T'}} \mathbb{E}_{q(x_{T'}|x_{S'})} \log p_{\theta}(X_{T'}|X_{S'})$$

where $q(S', T')$ denotes the sampling process of $(S', T')$ with random permutations, $T' = \{t_1, t_2, \cdots\}$, $\Xi'_{S', T'} = \{(S', t_1), (\{S', t_1\}, t_2), (\{S', t_1, t_2\}, t_3), \cdots\}$, often $p_{\theta}(X_{T'}|X_{S'}) = \text{Categorical}(X_{T'}; p_{\theta}(X_{S'}))$ is modeled as a categorical distribution with probabilities $p_{\theta}(X_{S'})$, and $X_{T'}$ always contain one token (either the label $y$ or an $x$-token). Refer also to Table 1 of the main manuscript for other details.

More results complementing Fig. 5 of the main manuscript are given in Fig. 11.
G Empirical evaluations on the GLUE benchmark

Concerning the empirical comparisons between existing methods for foundation models and the presented big learning, intuitively, one would consider first using the big learning as the pretraining strategy in place of existing ones, followed by applying the same naive fine-tuning on downstream tasks, to evaluate the effectiveness of the big learning. Unfortunately, we cannot afford the pretraining cost; for example, to pretrain a XLNet-Large takes about 5.5 days on 512 TPUs according to [56]. We leave that to the community, as mentioned in the Conclusion.

To demonstrate the advantages of the big learning over existing methods for foundation models, we alternatively consider leveraging it to serve as the less expensive fine-tuning strategy. It’s worth highlighting that, from another perspective, such experiments also verify the advantages of the big learning in the fields of supervised learning, when compared to existing supervised learning methods.

Specifically, we design experiments based on the Hugging Face transformers library [54], the GLUE benchmark [51], and the XLNET [56] that outperforms the BERT on many NLP tasks. We employ the same pretrained xlnet-base-cased model and continually train it on the downstream RTE/MRPC/SST-2 classification tasks via (i) the naive fine-tuning (i.e., identical to the original XLNET, termed FT) and (ii) the big learning (termed big-learn), respectively. In other words, the pretraining phase (i.e., the permutation language modeling [56], a special case of the big learning) is the same and we compare our big-learn with the naive FT during the finetuning phase.

Because the data of the downstream classification tasks contain both feature $x$ and label $y$, we resort to the big learning settings of Section 3.3 of the main manuscript. Specifically, $X = (y, x)$ and the universal foundation model $p_\theta(X_{T'}|X_{S'})$ has a network architecture similar to the one shown in Fig. 8 of the main manuscript. Note $p_\theta(X_{T'}|X_{S'})$ consists of the pretrained XLNET backbone and a task-specific head that is attached to the output of the $<CLS>$ token; for simplicity, we abuse $\theta$ to represent all the parameters. For a specific $(S', T')$ pair, $p_\theta(X_{T'}|X_{S'})$ recovers $p_\theta(y|x)$, i.e., a conventional classifier.

With the above notations, we next formalize the objective for both FT and our big-learn.

- **FT.** Often a cross-entropy is employed, which is identical to

$$L_{FT}(\theta) = E_{q_{\text{downstream}}(x,y)}[-\log p_\theta(y|x)], \quad (20)$$

where $q_{\text{downstream}}(x,y)$ represents the training data of the downstream classification task.

- **Big-learn.** For direct comparisons, we formalize the big-learn objective as

$$L_{\text{big-learn}}(\theta) = L_{FT}(\theta) + \beta_{\text{BigLearn}}L_{\theta}, \quad (21)$$

where $\beta_{\text{BigLearn}}$ is a hyperparameter and

$$L(\theta) = E_{q(S', T')}E_{q_{\text{downstream}}(x)}[-\log p_\theta(X_{T'}|X_{S'})], \quad (22)$$
with \( q(S', T') \) denoting the sampling process of \((S', T')\). We simply reuse the same sampling process in Table 3.

Note Eq. (22) is equivalent to minimizing \[ \mathbb{E}_{q(S', T')} \text{KL}[ q_{\text{downstream}}(X_T|X_S') || p_\theta(X_T|X_S') ] \] by referring to Eq. (1) of the main manuscript.

### Table 4: Tested hyperparameters when comparing FT with big-learn on the GLUE benchmark.

| Task  | Hyperparameter Learning Rate | #Epochs | WarmUp Steps | \( \beta_{\text{BigLearn}} \) |
|-------|-----------------------------|---------|--------------|-----------------------------|
| RTE   | [2e-5, 4e-5, 6e-5]          | [3, 4, 7, 10, 15] | [0, 120]       | [0., 0.2, 0.4, 0.6, 0.8] |
| MRPC  | [2e-5, 4e-5, 6e-5]          | [3, 4, 7, 10, 15] | [0, 120]       | [0., 0.2, 0.4, 0.6, 0.8] |
| SST-2 | [2e-5, 4e-5, 6e-5]          | [2, 3, 4]      | [0, 1200]      | [0., 0.2, 0.4]             |

We extensively compare FT with big-learn on the downstream RTE/MRPC/SST-2 classification tasks, by evaluating the accuracy and/or F1 score on the Dev set across the combinations of the tested hyperparameters shown in Table 4. The hyperparameters are chosen following [13, 56].

The best/median metrics are summarized in Table 4 and Fig. 12 shows the corresponding boxplots; it’s clear that our big-learn consistently outperforms FT. Accordingly, the big learning can serve as a superior fine-tuning strategy. It’s worth highlighting we did not carefully tune our big-learn; therefore, it’s likely that its performance could be further improved by e.g., tuning the sampling process \( q(S', T') \).

![Boxplots](image.png)

**Figure 12:** Boxplots of the Dev-set accuracies from FT and our big-learn. Note big-learn with \( \beta_{\text{BigLearn}} = 0 \) is identical to FT (see Eq. (21)). It’s clear that big-learn consistently outperforms FT on all three tasks.

We’d like to emphasize that the big learning can reduce the pretrain-finetuning gap because

- it can act as the pretraining and finetuning objectives, simultaneously;
- one can even rely on the big learning to completely merge the pretraining and finetuning phases, leading to a zero gap.
Motivated by the performance boost from the BERT to the XLNET and our discussions “on the generalization of model parameters and latent features” of Section 3.2 of the main manuscript, we posit that the big learning can serve as better pretraining and finetuning strategies than existing methods, leading to a universal machine learning paradigm. We leave the corresponding verification as future research.

**H Additional experimental results**

More experimental results, complementing the limited demonstrations of the main manuscript, are given below. Please refer to the captions for details.

Figure 13: Demonstrating the generation/completion capabilities of big learning when gradually increasing the ratio of \( S \) from 0 (joint generation) to 0.9, from left to right. Shown in the light-blue boxes of the first row are the masks of \( x_S \) applied in each column; white/black indicates \( S/T \). The right-most column shows ground-truth \( x \) shared in each row. Note each row also employs the same noise. It’s clear that the generations become increasingly similar/dissimilar to the ground-truth \( x \) as the ratio of \( S \) increases/decreases, as expected. See the category, style, and thickness of the MNIST generations as the ratio of \( S \) decreases, as well as the identity, expression, hairstyle, and gender of the CelebA generations. Big learning produces realistic and diverse generations/completions in all situations.
Figure 14: More MNIST generations/completions from big learning when gradually increasing the ratio of \( S \) from 0.0 to 0.9.
Figure 15: More CelebA generations/completions from big learning when gradually increasing the ratio of $S$ from 0.0 to 0.9.
Figure 16: The diverse generations/completions of big learning with (a)/(c) various $\mathcal{S}$ settings and (b)/(d) different noises. Shown in red boxes are either the ground-truth images $x$ or the source $x_0$. Big learning delivers diverse realistic generations w.r.t. different $\mathcal{S}$/noise settings.
Figure 17: The strong generalization capability of big learning w.r.t. anomalous testing cases out of the training domain. Big learning generalizes well on $x_{\tilde{s}}$ that are constructed with (a) random center patches replaced in the upper-left corner, (b) random center patches replaced in the upper part, (c) random center patches duplicated and replaced in the center, and (d) random patches and more complicated manipulations (including duplication, relocation, and mix-up).
Figure 18: The strong generalization capability of big learning w.r.t. anomalous/unseen testing cases out of the training domain, on (a) CelebA, (b) Flowers, and (c) MetFaces. Big learning generalizes well on $x_S$ constructed by (a) mixing-up patches from different CelebA images, (b) sampling out-of-domain image patches from the Flowers dataset, and (c) sampling out-of-domain image patches from the MetFaces dataset.
Figure 19: Out-of-domain generations/completions from big learning on the Flowers, when gradually increasing the ratio of $S$ from 0.0 to 0.9. The tested model is big-learned on the CelebA.
Figure 20: Out-of-domain generations/completions from big learning on the MetFaces, when gradually increasing the ratio of $S$ from 0.0 to 0.9. The tested model is big-learned on the CelebA.