Understanding Masked Autoencoders via Hierarchical Latent Variable Models

Lingjing Kong[*1]   Martin Q. Ma[*1]   Guangyi Chen[1,2]
Eric P. Xing[1,2]   Yuejie Chi[1]   Louis-Philippe Morency[†1]   Kun Zhang[†1,2]
[1]Carnegie Mellon University   [2]Mohamed bin Zayed University of Artificial Intelligence

Abstract

Masked autoencoder (MAE), a simple and effective self-supervised learning framework based on the reconstruction of masked image regions, has recently achieved prominent success in a variety of vision tasks. Despite the emergence of intriguing empirical observations on MAE, a theoretically principled understanding is still lacking. In this work, we formally characterize and justify existing empirical insights and provide theoretical guarantees of MAE. We formulate the underlying data-generating process as a hierarchical latent variable model and show that under reasonable assumptions, MAE provably identifies a set of latent variables in the hierarchical model, explaining why MAE can extract high-level information from pixels. Further, we show how key hyperparameters in MAE (the masking ratio and the patch size) determine which true latent variables to be recovered, therefore influencing the level of semantic information in the representation. Specifically, extremely large or small masking ratios inevitably lead to low-level representations. Our theory offers coherent explanations of existing empirical observations and provides insights for potential empirical improvements and fundamental limitations of the masking-reconstruction paradigm. We conduct extensive experiments to validate our theoretical insights.

1. Introduction

Self-supervised learning (SSL) has achieved tremendous success in learning transferable representations without labels, showing strong results in a variety of downstream tasks [12, 14, 16, 23, 49]. As a major SSL paradigm, masked image modeling (MIM) [1–3, 11, 13, 22, 41, 63, 69] performs the reconstruction of purposely masked image pixels as the pretraining task. Among MIM methods, masked autocoding (MAE) [22] has gained significant traction due to its computational efficiency and state-of-the-art performance in a wide range of downstream tasks.

Empirical observations from previous work reveal various intriguing properties of MAE. In particular, aggressive
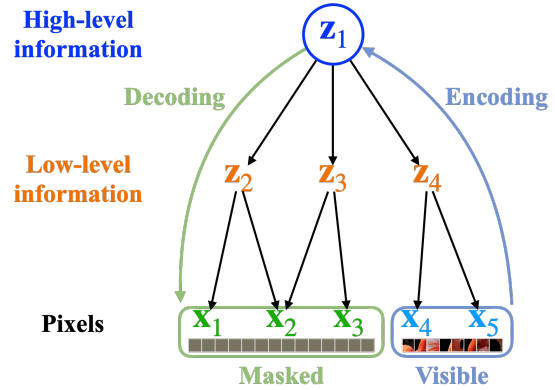

Figure 1. **Masking-reconstruction under a hierarchical generating process.** In a hierarchical data-generating process, high-level latent variables (e.g., z_1) represent high-level information such as semantics, and low-level latent variables (e.g., $[z_2, z_3, z_4]$) represent low-level information such as texture. We show that through proper masking, MAE learns to recover high-level latent variables with identifiability guarantees.

masking has been shown critical to downstream task performances [22, 28, 61, 63]. It is conjectured that such masking forces the model to learn meaningful *high-level* semantic understanding of the objects and scenes rather than the *low-level* information such as texture. However, it remains largely unclear whether such intuitions are sound in principle. Theoretically verifying and characterizing these empirical insights would not only grant a certificate to the current approaches but would also offer theoretical insights for algorithmic advancements.

In this work, we establish a principled yet intuitive framework for understanding MAE and providing identifiability guarantees. Concretely, we first formulate the underlying data-generating process as a hierarchical latent variable model (Figure 1), with high-level variables corresponding to abstract and semantic information like classes, and low-level variables corresponding to elaborate and granular information like texture. Such latent variable models have been studied in causal discovery [29, 62]. In [27, 50], it is hypothesized that complex data, such as images, follow a hierarchical latent structure.

Stemming from this formulation, we show that under reasonable assumptions, MAE can recover a subset of the

true latent variables within the hierarchy, where the levels of the learned latent variables are explicitly determined by how masking is performed. Our theoretical framework not only unifies existing empirical observations coherently but also gives rise to insights for potential empirical improvements and fundamental limitations of MAE. Our theory improves the existing nonlinear identifiability results [45, 58] and can be of independent interest.

Empirically, we deduce several insights from our theoretical results and verify them with experiments. Unlike common belief, MAE trained with extremely high masking ratios (e.g., 90%) captures low-level information, similar to models trained with extremely low ratios (e.g., 10%). Our results suggest that learning high-level semantic information is only possible in the non-extreme masking regime. We also discuss masking designs that can potentially improve current empirical performance.

Contributions. We highlight the following contributions:

- We formulate the underlying data-generating process as a hierarchical latent variable model. Under such a formulation, we provide a theoretical guarantee for MAE by showing that it can recover true latent variables in the hierarchical model.

- Based on our theoretical results, we establish the connection between masking hyperparameters (i.e., masking ratios and patch sizes) and the learned representation and discuss potential improvements and inherent limitations of MAE.

- We validate our theoretical insights with extensive experimental results. We illustrate how the semantic level of the learned representation varies with the aggressiveness of the masking strategy. Interestingly, representations learned under overly aggressive masking (e.g.,ß 90% masking ratio) exhibit similar properties to their counterparts learned with overly conservative masking (e.g., 10% masking ratio).

2. Theoretical Understanding

2.1. A Hierarchical Data-generating Process

Images, despite their high dimensionality, are well structured – there is a multitude of statistical dependencies among pixels determined by their relative distances and visual semantics. For instance, pixels in close proximity are often highly dependent, whereas pixels far apart typically share less information. There has been a plethora of work adopting this intuition for vision tasks such as image generation [47, 55, 67]. Similar insights are also addressed in attempts to learn a part-whole image representation [27, 50].
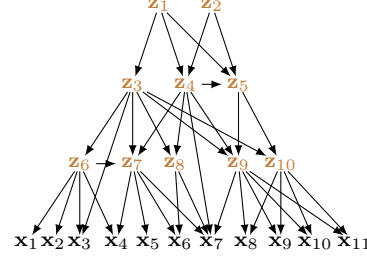

Figure 2. **A hierarchical data-generating process.** $\mathbf{z}$ represents the latent variables and $\mathbf{x}$ stands for the observable variables (i.e. image pixels). The hierarchical model is generic and is capable of modeling arbitrary DAGs in the latent space.

In this work, we formulate such an underlying structure of images with a hierarchical data-generating process [1, 29, 62] (Figure 2). Under this formulation, we reveal the underpinning principle of MAE and provide identifiability guarantees. In particular, we show that through masking-reconstruction, MAE learns the long-range statistical dependencies within the image, which renders it capable of extracting high-level semantic representations.

Formally, the generating process is defined with a graph structure $\mathbf{G} := (\mathbf{V}, \mathbf{E})$ where $\mathbf{E}$ is the set of all directed edges and $\mathbf{V} := (\mathbf{X}, \mathbf{Z})$ comprises all observable variables $\mathbf{X} := \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ (i.e., all pixels) and all latent variables $\mathbf{Z} := \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$. Each variable $\mathbf{x}_i$ or $\mathbf{z}_j$ represents a multidimensional vector. [1] The hierarchical latent structure $\mathbf{G}$ fulfills the following assumption:

Assumption 1. *(Data-generating process): There is no direct edge between any two observables:* $\forall \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$, $(\mathbf{x}_i, \mathbf{x}_j) \notin \mathbf{E}$ *and* $(\mathbf{x}_j, \mathbf{x}_i) \notin \mathbf{E}$. *Each variable is generated by its parents in a directed acyclic graph (DAG) according to:*

$$\begin{aligned} \mathbf{z}_i &= g_{\mathbf{z}_i}(Pa(\mathbf{z}_i), \boldsymbol{\varepsilon}_i), \\ \mathbf{x}_j &= g_{\mathbf{x}_j}(Pa(\mathbf{x}_j), \boldsymbol{\varepsilon}_j), \end{aligned} \quad (1)$$

where $g_{\mathbf{z}_i}$ *and* $g_{\mathbf{x}_j}$ *are invertible functions,* $\boldsymbol{\varepsilon}_i$ *denotes exogenous random variables, and* $Pa(\cdot)$ *denotes the parents of a certain node.*

The invertible data-generating-module assumption (g_i and g_j being invertible) is adopted from prior work identifying latent variables in deep generative models [18, 58]. We make the following remarks on the hierarchical generating process. First, we note that we impose minimal constraints on the graph structure among the latent variables (i.e., the connectivity among latent variables $\mathbf{z}$); therefore, the hierarchical model class is generic and encompasses all possible DAG structures over latent variables (Figure 2). Next, we interpret the latent variables $\mathbf{z}$ as information related to semantic/content information, such as the shape and contour

---

[1] In high-dimensional data like images, there is a larger degree of information redundancy, e.g., neighboring pixels. Thus, it is sensible to lump one-dimensional variables into vectors.

in the image, whereas the exogenous variables ε injected in each layer represent nuanced information, such as the texture and contrast of the image. Each structural function g_i mixes the two sources of information and generates a more low-level variable until pixels $\mathbf{x}$. Lastly, for the upcoming theoretical results, as long as the data-generating process conforms to the hierarchical graph assumption, our theory holds, and the insights do not rely on the knowledge of a specific graph structure.

2.2. Masked Autoencoders

As a canonical method of masking-reconstruction learning, MAE [22] randomly masks a subset of pixel patches in the original image and then reconstructs the masked patches from the encoded representation of the visible part. More formally, we formulate the MAE training as follows.

Mask sampling: random masks $\mathbf{m}$ are sampled from a distribution $p_{\mathbf{m}}$ which is parameterized by the masking ratio r (i.e., the ratio between the number of masked pixels and the number of all pixels) and patch size s (i.e., the size of the minimal masking unit).

MAE encoding: $E_{\mathbf{m}^c}(\mathbf{x}_{\mathbf{m}^c})$ maps the unmasked part $\mathbf{x}_{\mathbf{m}^c}$ to a latent representation $\hat{\mathbf{c}}$ [2], where $\mathbf{m}^c$ denotes the complement of the mask index set $\mathbf{m}$ and is passed to the encoder as positional embeddings to indicate the positions of the visible patches.

MAE decoding: $D_{\mathbf{m}}(\hat{\mathbf{c}}, \hat{\mathbf{s}}_{\mathbf{m}})$ reconstructs the masked image $\mathbf{x}_{\mathbf{m}}$ from the estimated latent variable $\hat{\mathbf{c}}$ (i.e., the encoder output), and the auxiliary information $\hat{\mathbf{s}}_{\mathbf{m}}$ embodying positional embeddings and [MASK] token which are fed to the decoder in MAE. Although $\hat{\mathbf{s}}_{\mathbf{m}}$ is deterministic in MAE implementation, we view it as a random variable in our analysis.

With the notation above, the MAE training objective can be expressed as follows:

$$L(E, D) := \mathbb{E}_{\mathbf{m}, \mathbf{x}, \hat{\mathbf{s}}_{\mathbf{m}}} \left[\| D_{\mathbf{m}} \left(E_{\mathbf{m}^c}(\mathbf{x}_{\mathbf{m}^c}), \hat{\mathbf{s}}_{\mathbf{m}} \right) - \mathbf{x}_{\mathbf{m}} \|^2 \right]. \quad (2)$$

2.3. Identifiability Theory

Building upon the formalization above, we show in Theorem 1 that each random mask $\mathbf{m}$ would induce a specific (sub)set of latent variables that fully captures the statistical dependency between the masked part and the visible part. We denote this relationship as $\mathbf{c} \subset \mathbf{Z}$ where $\mathbf{c}$ is the subset of the latent variable set $\mathbf{Z}$.

Theorem 1. *(Locating the shared information $\mathbf{c}$): In a hierarchical latent variable structure $\mathbf{G}$, for each specific mask $\mathbf{m}$, there exists a corresponding minimal set of latent variables $\mathbf{c}$ such that the generating process of $\mathbf{x}$ can be expressed as in Figure 3 where the following conditions are satisfied:*
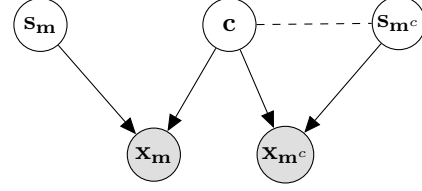
Figure 3. **Information sharing latent models**. Here, $\mathbf{x}_{\mathbf{m}}$ and $\mathbf{x}_{\mathbf{m}^c}$ denote the masked part and the visible part of the image $\mathbf{x}$, respectively. $\mathbf{c}$ stands for the maximally shared information between $\mathbf{x}_{\mathbf{m}}$ and $\mathbf{x}_{\mathbf{m}^c}$. $\mathbf{s}_{\mathbf{m}}$ and $\mathbf{s}_{\mathbf{m}^c}$ refer to the information specific to $\mathbf{x}_{\mathbf{m}}$ and $\mathbf{x}_{\mathbf{m}^c}$ respectively. The dashed line indicates the potential existence of statistical dependence.

1. $\mathbf{x}_{\mathbf{m}} = g_{\mathbf{x}_{\mathbf{m}}}(\mathbf{c}, \mathbf{s}_{\mathbf{m}})$ *and* $\mathbf{x}_{\mathbf{m}^c} = g_{\mathbf{x}_{\mathbf{m}^c}}(\mathbf{c}, \mathbf{s}_{\mathbf{m}^c})$ *where both $g_{\mathbf{x}_{\mathbf{m}}}$ and $g_{\mathbf{x}_{\mathbf{m}^c}}$ are invertible;*

2. $\mathbf{s}_{\mathbf{m}} \perp\!\!\!\perp (\mathbf{c}, \mathbf{s}_{\mathbf{m}^c})$*;*

3. $\mathbf{c}$ *is minimal:* $\forall \mathbf{c}' \subset \mathbf{Z}$ *such that $dim(\mathbf{c}') < dim(\mathbf{c})$, $\mathbf{c}'$ cannot satisfy the two conditions above.*

Such $\mathbf{c}$ and the corresponding $\mathbf{s}_{\mathbf{m}}$ are unique and can be located from the hierarchical structure by executing Algorithm 1. Furthermore, $\mathbf{s}_{\mathbf{m}^c}$ can be found through Algorithm 2.

The proof, Algorithm 1, and Algorithm 2 can be found in Appendix A. We note that although the minimal $\mathbf{c}$ and its corresponding $\mathbf{s}_{\mathbf{m}}$ are unique for a given mask $\mathbf{m}$, there is no unique $\mathbf{s}_{\mathbf{m}^c}$ in general. Algorithm 2 returns one such instance.

Theorem 1 states that for each mask $\mathbf{m}$, there exists a corresponding $\mathbf{c}$ that represents all the information contained in the visible part $\mathbf{x}_{\mathbf{m}^c}$ that is conducive to reconstructing the masked part $\mathbf{x}_{\mathbf{m}}$. Algorithm 1 can locate such $\mathbf{c}$ in the hierarchy and directly characterizes the impact of masking on the property of $\mathbf{c}$.

Next, in Theorem 2, we show that MAE learning objective (Equation 2) estimates $\mathbf{c}$ specified in Theorem 1, and MAE attains a form of identifiability of $\mathbf{c}$. We first lay out the assumptions:

Assumption 2. *(MAE model): For any mask $\mathbf{m}$, the MAE decoder $D_{\mathbf{m}}(\hat{\mathbf{c}}, \hat{\mathbf{s}}_{\mathbf{m}})$ has a non-singular Jacobian matrix almost anywhere, and there exists an invertible function $\tilde{g}_{\mathbf{m}^c}(\cdot)$ such that MAE encoder $E_{\mathbf{m}^c}(\cdot) = [\tilde{g}_{\mathbf{m}^c}^{-1}(\cdot)]_{1:d_c}$ where $[\cdot]_{1:d_c}$ denotes the dimensions corresponding to $\mathbf{c}$. Moreover, $(D_{\mathbf{m}}, \tilde{g}_{\mathbf{m}^c})$ forms an invertible mapping between $(\hat{\mathbf{c}}, \hat{\mathbf{s}}_{\mathbf{m}}, \hat{\mathbf{s}}_{\mathbf{m}^c})$ and $(\mathbf{x}_{\mathbf{m}}, \mathbf{x}_{\mathbf{m}^c})$*

Next, we show MAE identifies the shared information $\mathbf{c}$:

Theorem 2. *(Identifiability of $\mathbf{c}$): For each mask $\mathbf{m}$, given the dimensions $(d_{\mathbf{c}}, d_{\mathbf{s}_{\mathbf{m}}})$ the encoder function $E_{\mathbf{m}^c}(\cdot)$ recovers all information of $\mathbf{c}$ located in Theorem 1, i.e., there exists a one-to-one mapping h, s.t., $h(\mathbf{c}) = \hat{\mathbf{c}}$.*

In the following, we discuss our assumptions and results. The proof can be found in Appendix B.

Assumption interpretation. Assumption 1 follows prior work identifying latent variables in deep generative models [18, 58] to ensure that latent variables are recoverable from pixels. Assumption 2 requires the MAE encoder $E_{\mathbf{m}^c}$ to be part of an invertible function output – this is mild and allows the encoder to be more flexible than invertible functions. The decoder $D_{\mathbf{m}}(\hat{\mathbf{c}}, \hat{\mathbf{s}}_{\mathbf{m}})$ is assumed to be locally invertible in $\hat{\mathbf{c}}$ almost surely, allowing for a broader class than invertible functions, e.g., nondegenerate polynomials. The joint invertibility of $(D_{\mathbf{m}}, \tilde{g}_{\mathbf{m}^c})$ ensures no information loss during the estimation process.

How does MAE work? Theorem 2 states that the MAE objective (Equation 2) essentially serves to estimate the shared variable $\mathbf{c}$ and is able to restore all information in $\mathbf{c}$. Therefore, the efficacy of MAE stems from its ability to extract high-level semantic representations from low-level features like image pixels. Moreover, our theory indicates the possibility of fully identifying a latent hierarchical structure via properly designed self-supervised objectives, opening up research avenues for future work.

Takeaway: *MAE provably recovers high-level representations from low-level features like pixels.*

How does masking influence the learned representation? Theorem 1 establishes a direct connection between the mask $\mathbf{m}$ and the shared information $\mathbf{c}$, which is further connected to the MAE estimate $\hat{\mathbf{c}}$ in Theorem 2. We can observe that conservative masking with overly small masking ratios and masking patch sizes inevitably leads to low-level latent variables. To see this, in Figure 4a, the mask is not large enough to cover all observable descendants of a desirable high-level variable $\mathbf{z}_1$, thus following Algorithm 1 a low-level variable $\mathbf{z}_3$ will mix in $\hat{\mathbf{c}}$, preventing the model from learning $\mathbf{z}_1$. This insight highlights the necessity of nontrivial masking ratios and patch sizes and resonates with the empirical observations in [22, 28, 63].

Surprisingly, the above reasoning can be applied to the case with extremely aggressive masking: in Figure 4b low-level latent variables $\mathbf{z}_6$ will be learned by MAE when the visible part is too small to cover all observable descendants of a desirable high-level variable $\mathbf{z}_2$. Thus, the learned representation does not become monotonically more high-level with increasing masking aggressiveness – overly aggressive masking also gives rise to low-level representations. This insight echoes the empirical finding in [61, 63] where the extremely large masking degrades the performance of high-level downstream tasks like classification [63] but yields relatively low-level representations like the object loca-
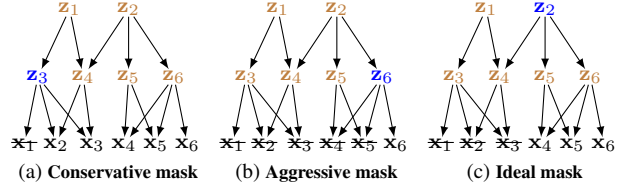

Figure 4. **The impact of masking on the learned representation.** We label the masked pixels with $\mathbf{x}$. We locate the MAE learned latent variables with Algorithm 1 and label them with blue. We can observe that extremely low (left) and high (middle) masking intensities lead to low-level representations, whereas the desirable masking intensity that yields a high-level representation lies in the intermediate masking aggressiveness.

tions/scales in the image [61]. In Section 3, we present empirical evidence to verify our theoretical insights.

Takeaway: *(1) MAE under different masking intensities learns representations of different abstraction levels; (2) Learning high-level representations is very hard with extreme masking.*

Is current MAE optimal for representation learning? As reflected in the discussion above, although MAE offers the flexibility of tuning the masking scheme to learn representations of various levels, it is inherently challenging to learn high-level representations by random masking without prior knowledge of the latent structure. In contrast, contrastive learning [5,9,10,12,14,23,64] actively leverages the prior knowledge encoded in data augmentations to extract the augmentation-invariant latent variables [58] which correspond to the high-level latent variables in our hierarchical model. Our theory suggests an explanation for why representations learned by contrastive learning are superior to those of MAE on high-level tasks like linear-probing classification.

Takeaway: *Learning high-level representations can be challenging for random masking.*

3. Experiments

We conduct five sets of experiments and then provide insights into possible empirical improvements over MAE. We investigate the following question: *how does the masking aggressiveness influence the representation?* To this end, we pretrain MAE using different masking ratios and making patch sizes, and then conduct the following evaluations: 1) measuring structure-level and pixel-level similarities between the reconstructed and the original images; 2) visualizing self-attentions to understand what is learned; 3) performing linear probing on ImageNet-1K (IN1K) and different ImageNet variants; 4) measuring the shape bias [19] which estimates how much a network leverages high-level shape information over low-level texture information; and
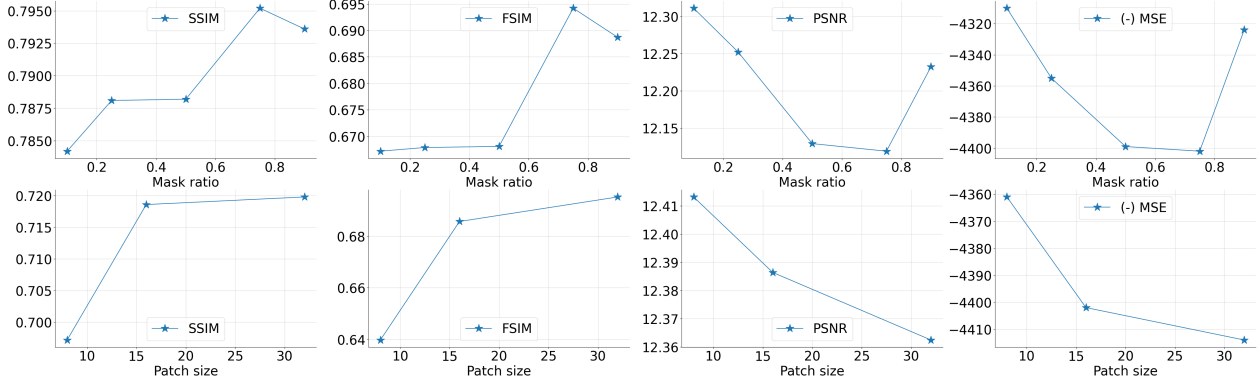
Figure 5. **Reconstruction evaluation** using the validation set without masking, based on two structural-level similarity metrics (SSIM and FSIM) and two pixel-level metrics (PSNR and MSE). We plot negative MSE for easier visualization. Higher SSIM and FSIM indicate high-level information is better captured, while higher PSNR and negative MSE indicates better low-level reconstruction.

5) transfer learning on object detection and segmentation. Details of experiments can be found in Appendix.

Pretraining overview. We conduct pretraining on IN1K using the MAE pipeline [22], with ViT-Base as the backbone of our study. We conduct two sets of pretraining: 1) fixing patch size at 16 and varying the masking ratios from $\{0.1, 0.25, 0.5, 0.75, 0.9\}$. Larger masking ratios suggest larger portions of pixels being masked, i.e., 0.9 suggests 90% of pixels being randomly masked for the encoder. 2) Fix the masking ratio at 0.75 and vary the patch size from $\{8, 16, 32\}$. To decouple the patch size for masking images and the patch size hyperparameter in the Vision Transformer, we adopt the implementation from [28]. The patch size studied in this paper refers to the minimal *masking unit* size, and the hyperparameter of the ViT patch size remains fixed at 8.

3.1. Reconstructing High-level or Low-level Representations

Setup. We begin our study by evaluating the high-level structural and low-level pixel-wise similarities between the reconstructed images from MAE and the original inputs. We choose two metrics for high-level similarities and two metrics for low-level similarities. If the structural similarities are high, MAE captures more perceivable structural semantics from the input. The two high-level similarities are structural similarity index measure [60] (**SSIM**) and feature similarity index measure [65] (**FSIM**). Both metrics consider the change of perceptions in structural information [33]. SSIM considers the normalized mean value of the structural similarity between the original and reconstructed images, and FSIM considers the normalized mean value of the feature similarity between the two images. A higher SSIM or a higher FSIM suggests a better reconstruction of high-level information (structural or feature-wise). On the other hand, if the pixel-level similarity between recon-

structed images and the original input is high, then MAE is deemed to capture the low-level information about the input better. The two low-level metrics are the mean squared error (**MSE**), which is the squared differences between the original and reconstructed images in the pixel space, and the peak signal-to-noise ratio (**PSNR**), which measures the ratio between the power of the maximum possible pixel value and the power of corruption noise. A lower MSE or a *higher* PSNR suggests a better reconstruction at the pixel level. Note that a very low MSE or a very high PSNR may also suggest that the model captures high-level information well. All four metrics are full reference, meaning the assessment is based on comparing original and reconstructed images rather than the reconstructed output. We introduce the high-level and low-level metrics below and perform the reconstructions on the IN1K evaluation set. The full details and comparisons of the four metrics can be found in [51].

Evaluation of image reconstructions. We include the results in Figure 5. We plot the negative of the MSE to show a consistent trend with PSNR, so higher means better low-level reconstruction. From the first row, varying masking ratios from 0.1 to 0.75, higher masking ratios produce reconstructions with higher structural information similarities with the original image (higher SSIM and FSIM), but the model trained with the extremely high ratio 0.9 captures more low-level information (higher PSNR and higher negative MSE). On the other hand, lower masking ratios tend to reconstruct images that capture low-level information better. From the second row, larger patch sizes produce image reconstructions that capture high-level similarities better, while smaller patch sizes have low-level metrics. The empirical observations validate our insight from Section 2.3: *higher masking ratios and patch sizes capture high-level structural information better*, but *extreme masking ratios (both low and high) capture less high-level and more low-level information.*
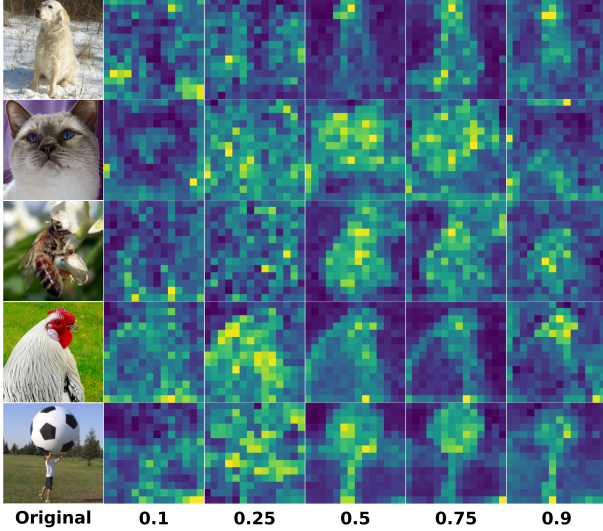
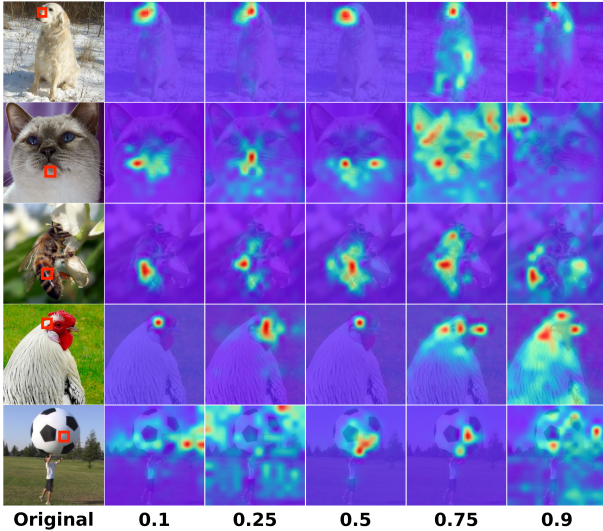Figure 6. **Self-attention of the `[CLS]` tokens** averaged across the heads of the last layer in MAE.

Figure 7. **Self-attention of an object-related token**. Chosen tokens are shown in red squares: dog nose, cat chin, bee abdomen, chicken head, and football center, respectively.

3.2. Attention Analysis

In this section, we measure the property of the learned representations of MAE by probing the attention heads. We would like to understand visually how masking ratios and patch sizes influence MAE's capacity to capture object-centric semantics. We provide two types of visualization: self-attention on the `[CLS]` token and self-attention on an object-related token. `[CLS]` has been considered a compact token to represent the whole image for downstream tasks, although recent work [22] suggests that the average pooling of all tokens may achieve slightly better results. Therefore, we also provide an analysis of object-related tokens to evaluate if MAE can contextualize object informa-

tion across tokens.

We plot examples of self-attention of the `[CLS]` token in Figure 6 and self-attention of non-CLS tokens related to the object in Figure 7. From the visualizations, as the masking ratio increases from 10% to 90%, the model is increasingly more able to grasp succinct information about the holistic objects rather than only focusing on the regions around the chosen token. However, extreme ratio 0.9 contains more low-level information and background information and cannot capture most of the remaining tokens related to objects (e.g., the dog, cat, and bee images in Figure 7). Extremely low masking ratios such as 0.1 capture both object-related and background tokens. Similarly, extreme masking ratios contextualize over other object-related tokens worse than intermediate masking ratios. We include the visualizations for patch sizes in Appendix. We observe that models trained with larger patch sizes better capture high-level information, but extreme patch size hurts, which validates our theoretical insight that *moderate masking ratios and patch sizes are critical for MAE to learn succinct and comprehensive object information.*

3.3. Representation Linear Separability

T-SNE embedding visualizations. To gain a visual understanding of how masking ratios and patch sizes influence the representation structure, we visualize T-SNE [57] embeddings of different models. We randomly select ten classes from ImageNet. The results are shown in Figure 8. From 0.1 to 0.75, a larger masking ratio consistently produces a more linearly separable representation, while the linear separabilities of representations with masking ratios 0.75 and 0.9 are visually similar. For different patch sizes, the embeddings are more separated as the patch sizes grow. *Non-extreme masking ratios and larger patch sizes generate more linearly separable embeddings.*

Linear probing on IN1K. We use linear probing to test how linearly separable the features are in the learned MAE representation. We show the linear probing results in Table 1 in row 1N1K. For different masking ratios, similar to the observation in [22], the accuracy increases steadily until the masking ratio reaches the sweet point of 0.75. An extremely large masking ratio (0.9) hurts performance. For different patch sizes, which are not shown in [22], we observe that the accuracy increases first from 8 to 16, then decreases significantly when the patch size is 32. From the results, higher masking ratios and larger patch sizes perform better at linear probing than lower masking ratios, but extreme masking hurts linear probing.

Robustness evaluation on ImageNet variants. We evaluate the robustness of the MAE models on different variants of ImageNet validation datasets, or object detection datasets
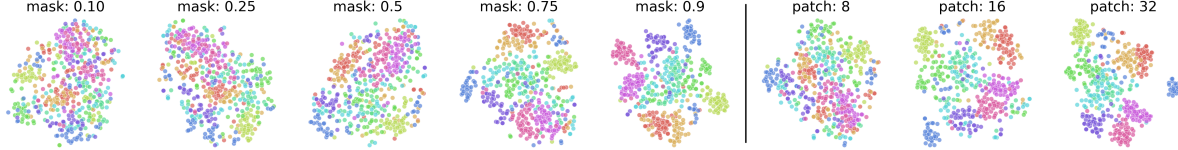
Figure 8. T-SNE embeddings of different MAE models under varied masking ratios and patch sizes. We fix the patch size at 16 to vary the masking ratios and fix the masking ratio at 0.75 to change the patch sizes. Each color represents one ImageNet class.

mask ratio	patch size	IN1K	IN-v2	OJN	IN-R	IN-A	IN-S
0.1	16	47.45	34.72	9.42	14.63	2.00	7.25
0.25	16	53.58	40.34	11.54	18.68	2.49	10.27
0.5	16	60.07	46.71	13.94	22.44	2.89	12.58
0.75	16	67.41	54.23	18.24	25.20	3.76	15.51
0.9	16	62.97	49.52	15.87	19.11	2.76	10.46
0.75	8	62.57	49.17	13.44	19.42	3.73	10.73
0.75	16	68.96	55.94	13.73	24.23	6.29	18.81
0.75	32	73.31	61.35	19.03	27.84	12.69	28.30

Table 1. **Accuracy (%) of linear probing and robustness evaluation** on ImageNet variants and ObjectNet. We linear-probe MAE via supervised training on IN1K, and then perform inference on IN1K as well as other evaluation sets.

that share similar class information with ImageNet-1K: ImageNet-v2 (INV2) [52], ObjectNet (OJN) [4], ImageNet-Adversarial (IN-A) [25], ImageNet-Rendition [4], and ImageNet-Sketch (IN-S) [59]. These datasets share similar semantics and labels with ImageNet but are under different data distributions. The MAE models are first trained in a supervised fashion on IN1K for linear probing, and inference is run on the evaluation sets without any training. Table 1 shows for all evaluation datasets, a reasonably large masking ratio (i.e., 0.75) achieves better robustness than smaller (i.e., 0.25) masking ratios, although extremely large (0.9) or small (0.1) masking ratios hurt the performance. For patch sizes, larger patch sizes yield better robustness evaluations on IN-v2, OJN, IN-R, and IN-S. *Non-extreme masking ratios and large patch sizes have stronger robustness performances than extreme masking ratios or patch sizes.*

3.4. Shape Bias

Texture vs. shape bias. Next, we analyze to what extent different MAE models rely on high-level vs. low-level information. We follow the analysis in [19], where the authors study whether a model leverages more low-level textures than high-level shapes for classification. As shown in Table 2, intermediate masking ratios (i.e., $0.25, 0.5,$ and 0.75) show a high level of shape bias, suggesting that the corresponding models exploit more high-level shape information. In contrast, extreme masking ratios (i.e., 0.1 and 0.9) leverage more low-level textures. This suggests that *extreme masking schemes make it more difficult to capture high-level shapes for MAE.*

3.5. Transfer Learning

Next, we evaluate the quality of MAE models on different downstream tasks. Specifically, we look at object de-

mask ratio	0.1	0.25	0.5	0.75	0.9
shape bias	0.1352	0.2545	0.2458	0.2563	0.2014

Table 2. **Shape bias** [19] measurement, a higher metric indicates that the model classifies images relying on the high-level shape feature rather than the low-level texture feature.

mask ratio	mask size	AP^{box}	AP^{mask}
0.1	16	30.47	28.24
0.25	16	32.38	29.95
0.5	16	34.87	32.11
0.75	16	**39.72**	**36.35**
0.9	16	37.17	34.35

Table 3. **COCO object detection and segmentation** using a ViT Mask R-CNN baseline.

tection and segmentation on the COCO dataset [43], which requires a strong semantic understanding of the scenes. We finetune Mask R-CNN [24] end-to-end using MAE-pretrained ViT weights. Following the practice in [22], we adapt the ViT backbone to make it compatible with FPN [42]. In Table 3, we report box AP for object detection and mask AP for instance segmentation. We reduce the number of epochs to 45 due to computational constraints. We observe that the 0.75 masking ratio yields the best detection and segmentation average precision, suggesting that the masking ratio 0.75 generates representation with the best semantic understanding. The extremely high masking ratio of 0.9 and a low masking ratio of 0.1 hurt the performance. Results of different patch size experiments are included in Appendix. The results suggest that *higher, but not extreme, masking ratios generate the best representation of object detection and segmentation tasks.*

3.6. Potential Algorithmic Improvements

Lastly, we discuss empirical suggestions based on our results that could benefit the performance of MAE.

First, as discussed in Section 2, when reconstructing the masked pixels near the boundary between the masked and unmasked regions, the model uses nearby visible pixels to interpolate, therefore capturing low-level pixel information. If high-level representation is desired for downstream tasks, the boundary pixels may be ignored when calculating the objective function.

Next, in light of the limitation of random masking in Section 2, one may leverage the latent structure of the underlying data-generating process for masking designs, which can serve as a more principled approach than recent work that

exploits auxiliary information for masking [34, 40, 41, 53]. To this end, one may take advantage of the recent development of causal discovery [29, 62] to identify the latent structure.

Lastly, if low-level information is preferable for downstream tasks, an extremely high masking ratio can retain such information and is more computationally efficient than its low masking ratio counterpart.

4. Related work

4.1. Masked Autoencoders

Masked image modeling (MIM) [1–3, 11, 13, 22, 41, 63, 69] has been gaining momentum recently due to their sota-of-the-art performances over many downstream tasks. The pretraining objective is simple in its basic form: the model is tasked to predict the masked-out image pixels with the information of the unmasked part. Despite the simplicity of the task, many intriguing properties have been observed on MIM that escape rigorous analysis. For instance, small masking ratios and masking patch sizes are empirically shown detrimental to downstream tasks like classification [22, 28]. It is hypothesized that aggressive masking forces to model to leverage more global information, rather than local interpolation [22]. However, whether such intuition is theoretically justifiable remains elusive. In this work, we provide theoretical verification of such intuitions and further derive insights into MAE's empirical behavior.

4.2. Theoretical Understanding of MAE

Despite the prominent success of MAE, only a limited number of papers are dedicated to understanding its underlying mechanism in a principled manner [8, 39, 48, 66]. Lee et al. [39] establish the connection between the inpainting pretraining task and downstream tasks by assuming that the downstream task target captures the statistical dependency between the visible part and the masked part in the inpainting. Under this assumption, they show that the sampling complexity of the downstream task can be largely reduced by pretraining. Cao et al. [8] inquire into the interactions between the transformer architecture and the MAE representation, highlighting the critical role of the attention mechanism in the success of MAE. Pan et al. [48] make a multi-view assumption on the samples, showing that MAE can extract class-relevant semantics with shallow convolutional models. Zhang et al. [66] study masking through the data-augmentation perspective and employ the augmentation graph [21] to illustrate the impact of masking on downstream task performance. In contrast, our work employs the hierarchical latent variable model, which lets us directly examine the relationship between the masking operation and the learned representations. Also, our theoretical guarantee is on the statistical identifiability of the true data-generating process rather than the statistical/optimization complexities as in most prior work.

4.3. Identifiability Guarantees for Nonlinear Latent-variable Models

In unsupervised learning, identifiability means latent variables involved in the underlying data-generating process can be estimated from observational data. This is critical to tasks like feature disentanglement [7, 26, 30, 35, 62] in the image generation community. However, principled disentanglement in the non-linear regime is challenging and even proved impossible without additional assumptions on the data-generating process [44]. Recent advances in independent component analysis (ICA) [6, 15, 31] obtain identifiability in the non-linear regime by imposing additional constraints on either the latent variable distribution or the function class variables [20, 32, 36–38, 45, 54, 58, 68]. Most relevant to ours are the identifiability theories in [45, 58] in which similar latent causal models (Figure 3) are studied. Specifically, our model allows the generating functions $g_{\mathbf{m}} \neq g_{\mathbf{m}^c}$ to be distinct (cf. identical functions assumed in [58]) and statistical dependence between $\mathbf{c}$ and $\mathbf{s}_{\mathbf{m}^c}$ (cf. independence assumed in [46]). Additionally, both works [46, 58] focus on contrastive learning with data augmentation, while our subject is MAE.

5. Conclusion

In this work, we formulate the data-generating process as a hierarchical latent variable model and provide guarantees that MAE can identify the true variables in such a hierarchical latent model. We then show how different masking ratios and patch sizes determine the set of true latent variables to be recovered, which influences the representation abstractions learned in MAE. Empirically, we show that non-extreme masking ratios or patch sizes often capture succinct and robust high-level information, while extreme masking ratios capture more low-level information.

References

[1] Animashree Anandkumar, Daniel Hsu, Adel Javanmard, and Sham Kakade. Learning linear bayesian networks with latent variables. In *International Conference on Machine Learning*, pages 249–257. PMLR, 2013. 1, 2, 8

[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. 1, 8

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 1, 8

[4] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019. 7, 15

[5] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 4

[6] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995. 8

[7] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta vae. *arXiv preprint arXiv:1804.03599*, 2018. 8

[8] Shuhao Cao, Peng Xu, and David A. Clifton. How to understand masked autoencoders, 2022. 8

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *ArXiv*, abs/2006.09882, 2020. 4

[10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 4, 15

[11] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pre-training from pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020. 1, 8

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 4

[13] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022. 1, 8

[14] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021. 1, 4

[15] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994. 8

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[17] 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 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 14

[18] Locatello et al. Weakly-supervised disentanglement without compromises. 2, 4

[19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 4, 7, 15

[20] Hermanni Hälvä and Aapo Hyvarinen. Hidden markov nonlinear ica: Unsupervised learning from nonstationary time series. In *Conference on Uncertainty in Artificial Intelligence*, pages 939–948. PMLR, 2020. 8

[21] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021. 8

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 1, 3, 4, 5, 6, 7, 8, 14, 16

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1, 4

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 7, 16

[25] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 7, 15

[26] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. 8

[27] Geoffrey Hinton. How to represent part-whole hierarchies in a neural network. *arXiv preprint arXiv:2102.12627*, 2021. 1, 2

[28] Ronghang Hu, Shoubhik Debnath, Saining Xie, and Xinlei Chen. Exploring long-sequence masked autoencoders. *arXiv preprint arXiv:2210.07224*, 2022. 1, 4, 5, 8, 14

[29] Biwei Huang, Charles Jia Han Low, Feng Xie, Clark Glymour, and Kun Zhang. Latent hierarchical causal structure discovery with rank constraints. *arXiv preprint arXiv:2210.01798*, 2022. 1, 2, 8

[30] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018. 8

[31] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, Inc, 2001. 8

[32] Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019. 8

[33] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5

[34] Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. *arXiv preprint arXiv:2203.12719*, 2022. 8

[35] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 8

[36] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020. 8

[37] Lingjing Kong, Shaoan Xie, Weiran Yao, Yujia Zheng, Guangyi Chen, Petar Stojanov, Victor Akinwande, and Kun Zhang. Partial disentanglement for domain adaptation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11455–11472. PMLR, 17–23 Jul 2022. 8

[38] Sébastien Lachapelle, Pau Rodríguez López, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ica. *arXiv preprint arXiv:2107.10098*, 2021. 8

[39] Jason D. Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting what you already know helps: Provable self-supervised learning, 2021. 8

[40] Gang Li, Heliang Zheng, Daqing Liu, Chaoyue Wang, Bing Su, and Changwen Zheng. Semmae: Semantic-guided masking for learning masked autoencoders. *arXiv preprint arXiv:2206.10207*, 2022. 8

[41] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. *Advances in Neural Information Processing Systems*, 34:13165–13176, 2021. 1, 8

[42] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 7, 16

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7, 16

[44] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019. 8

[45] Qi Lyu, Xiao Fu, Weiran Wang, and Songtao Lu. Latent correlation-based multiview learning and self-supervision: A unifying perspective. *arXiv preprint arXiv:2106.07115*, 2021. 2, 8

[46] Qi Lyu, Xiao Fu, Weiran Wang, and Songtao Lu. Understanding latent correlation-based multiview learning and self-supervision: An identifiability perspective. In *International Conference on Learning Representations*, 2022. 8

[47] Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. *Advances in neural information processing systems*, 32, 2019. 2

[48] Jiachun Pan, Pan Zhou, and Shuicheng Yan. Towards understanding why mask-reconstruction pretraining helps in downstream tasks. *arXiv preprint arXiv:2206.03826*, 2022. 8

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1

[50] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017. 1, 2

[51] Umme Sara, Morium Akter, and Mohammad Shorif Uddin. Image quality assessment through fsim, ssim, mse and psnr—a comparative study. *Journal of Computer and Communications*, 7(3):8–18, 2019. 5

[52] Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on imagenet. In *International Conference on Machine Learning*, pages 8634–8644. PMLR, 2020. 7, 15

[53] Yuge Shi, N. Siddharth, Philip H. S. Torr, and Adam R. Kosiorek. Adversarial masking for self-supervised learning, 2022. 8

[54] Peter Sorrenson, Carsten Rother, and Ullrich Köthe. Disentanglement by nonlinear ica with general incompressible-

flow networks (gin). *arXiv preprint arXiv:2001.04872*, 2020. 8

[55] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020. 2

[56] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245, 2014. 15

[57] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 6

[58] Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021. 2, 4, 8

[59] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019. 7, 15

[60] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[61] Zhirong Wu, Zihang Lai, Xiao Sun, and Stephen Lin. Extreme masking for learning instance and distributed visual representations. *arXiv preprint arXiv:2206.04667*, 2022. 1, 4

[62] Feng Xie, Biwei Huang, Zhengming Chen, Yangbo He, Zhi Geng, and Kun Zhang. Identification of linear non-gaussian latent hierarchical structure. In *International Conference on Machine Learning*, pages 24370–24387. PMLR, 2022. 1, 2, 8

[63] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 1, 4, 8

[64] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 4

[65] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011. 5

[66] Qi Zhang, Yifei Wang, and Yisen Wang. How mask matters: Towards theoretical understandings of masked autoencoders. *arXiv preprint arXiv:2210.08344*, 2022. 8

[67] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from generative models. *arXiv preprint arXiv:1702.08396*, 2017. 2

[68] Yujia Zheng, Ignavier Ng, and Kun Zhang. On the identifiability of nonlinear ica: Sparsity and beyond. *arXiv preprint arXiv:2206.07751*, 2022. 8

[69] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 1, 8

A. Proof for Theorem 1

In this section, we provide the proof for Theorem 1.

Theorem 1. *(Locating the shared information* $\mathbf{c}$*): In a hierarchical latent variable structure* $\mathbf{G}$*, for each specific mask* $\mathbf{m}$*, there exists a corresponding minimal set of latent variables* $\mathbf{c}$ *such that the generating process of* $\mathbf{x}$ *can be expressed as in Figure 3 where the following conditions are satisfied:*

1. $\mathbf{x_m} = g_{\mathbf{x_m}}(\mathbf{c}, \mathbf{s_m})$ *and* $\mathbf{x_{m^c}} = g_{\mathbf{x_{m^c}}}(\mathbf{c}, \mathbf{s_{m^c}})$ *where both* $g_{\mathbf{x_m}}$ *and* $g_{\mathbf{x_{m^c}}}$ *are invertible;*

2. $\mathbf{s_m} \perp\!\!\!\perp (\mathbf{c}, \mathbf{s_{m^c}})$*;*

3. $\mathbf{c}$ *is minimal:* $\forall \mathbf{c}' \subset \mathbf{Z}$ *such that* $dim(\mathbf{c}') < dim(\mathbf{c})$*,* $\mathbf{c}'$ *cannot satisfy the two conditions above.*

Such $\mathbf{c}$ *and the corresponding* $\mathbf{s_m}$ *are unique and can be located from the hierarchical structure by executing Algorithm 1. Furthermore,* $\mathbf{s_{m^c}}$ *can be found through Algorithm 2.*

Algorithm 1 Search for the minimal $\mathbf{c}$ and $\mathbf{s_m}$. $\mathbf{c}$ and $\mathbf{s_m}$ discussed in text can be viewed as the concatenations of vectors in $\mathcal{C}$ and $\mathcal{S_m}$. LocateParents($\cdot$) pins down the locations $\mathcal{Z}$'s parents (including exogenous variables) in the graph. DirectedPaths($\mathbf{d}, \mathcal{X}_{\mathbf{m}}^c$) returns the set of variables on the directed paths between $\mathbf{d}$ and $\mathcal{X}_{\mathbf{m}}^c$.

1: **inputs**: The hierarchical graph structure $\mathbf{G}$, and the partitioned observables $\mathcal{X_m}, \mathcal{X_{m^c}}$.
2: $\mathcal{C}, \mathcal{S_m} \leftarrow \emptyset, \emptyset$.
3: **Selection stage:**
4: **for** $\mathbf{x} \in \mathcal{X_m}$ **do**
5: $\quad \mathcal{Z} \leftarrow \{\mathbf{x}\}$.
6: $\quad$ **while** $\mathcal{Z} \neq \emptyset$ **do**
7: $\quad\quad \mathcal{Z}, \mathcal{E} \leftarrow$ LocateParents($\mathcal{Z}$)
8: $\quad\quad \mathcal{S_m} \leftarrow \mathcal{S_m} \cup \mathcal{E}$
9: $\quad\quad$ **for** $\mathbf{p} \in \mathcal{Z}$ **do**
10: $\quad\quad\quad$ **if** $\mathbf{p} \in$ Ancestors($\mathbf{x_{m^c}}$) **then**
11: $\quad\quad\quad\quad \mathcal{C} \leftarrow \mathcal{C} \cup \{\mathbf{p}\}$
12: $\quad\quad\quad\quad \mathcal{Z} \leftarrow \mathcal{Z} \setminus \{\mathbf{p}\}$
13: **Pruning stage:**
14: **for** $\mathbf{d} \in \mathcal{C}$ **do**
15: $\quad$ **for** $\mathbf{d}' \in \mathcal{C} \setminus \{\mathbf{d}\}$ **do**
16: $\quad\quad$ **if** $\mathbf{d}' \in$ DirectedPaths($\mathbf{d}, \mathcal{X_{m^c}}$) **then**
17: $\quad\quad\quad \mathcal{C} \leftarrow \mathcal{C} \setminus \{\mathbf{d}\}$
$\quad$ **return** $\mathcal{C}, \mathcal{S_m}$

Algorithm 2 Search for $\mathbf{s_{m^c}}$ **given** $\mathcal{C}$. LocateParents($\mathbf{p}$) pins down the locations $\mathbf{p}$'s parents (including exogenous variables) in the graph.

1: **inputs**: The hierarchical graph structure $\mathbf{G}$, the partitioned observables $\mathcal{X_m}, \mathcal{X_{m^c}}$, and $\mathcal{C}$ returned by Algorithm 1.
2: $\mathcal{S_{m^c}} \leftarrow \emptyset$.
3: **for** $\mathbf{x} \in \mathcal{X_{m^c}}$ **do**
4: $\quad \mathcal{P}, \mathcal{P}' \leftarrow \{\mathbf{x}\}, \emptyset$.
5: $\quad$ **while** $\mathcal{P} \neq \emptyset$ **do**
6: $\quad\quad$ **for** $\mathbf{p} \in \mathcal{P}$ **do**
7: $\quad\quad\quad$ **for** $\mathbf{p}' \in$ LocateParents($\mathbf{p}$) **do**
8: $\quad\quad\quad\quad$ **if** $\mathbf{p}'$ is exogenous **then**
9: $\quad\quad\quad\quad\quad \mathcal{S_{m^c}} \leftarrow \mathcal{S_{m^c}} \cup \{\mathbf{p}'\}$
10: $\quad\quad\quad\quad$ **else if** $\mathbf{p}' \in \mathcal{C}$ **then**
11: $\quad\quad\quad\quad\quad \mathcal{S_{m^c}} \leftarrow \mathcal{S_{m^c}} \cup ($LocateParents($\mathbf{p}$) $\setminus \{\mathbf{p}'\})$
12: $\quad\quad\quad\quad$ **else**
13: $\quad\quad\quad\quad\quad \mathcal{P}' \leftarrow \mathcal{P}' \cup \{\mathbf{p}'\}$
14: $\quad\quad \mathcal{P} \leftarrow \mathcal{P}'$
$\quad$ **return** $\mathcal{S_{m^c}}$

Proof. We will show that Algorithm 1 returns the minimal set of variables that satisfy all conditions in Theorem 1, which implies its existence. We will then argue that such $\mathcal{C}$ is unique for a specific mask $\mathbf{m}$.

Condition 1: We first discuss the invertibility of $g_{\mathbf{x_m}}$. Due to the invertibility assumption of the generating process, each backtrack step in Algorithm 1 is invertible (lossless). Thus, before the pruning stage, the mapping between $(\mathcal{C}, \mathcal{S_m})$ and $\mathcal{X_m}$ is invertible, as the information of $\mathcal{X_m}$ is either stored in either $\mathcal{C}$ or $\mathcal{S_m}$. We now show that the pruning stage does not break this invertibility. To see this, we note that for each $\mathbf{c}$ that is removed in the pruning stage, there exists $\mathbf{c}' \in \mathcal{C}$ on the directed path from $\mathbf{c}$ to $\mathcal{X_{m^c}}$ (per Algorithm 1). Therefore, $\mathbf{c}$ is a parent/ancestor of $\mathbf{c}'$ and can thus be retrieved by backtracking from $\mathbf{c}'$ thanks to the invertibility of the generating process. Therefore, the mapping between $(\mathcal{C}, \mathcal{S_m})$ and $\mathcal{X_m}$ is invertible.

We now address the invertibility of $g_{\mathbf{x_{m^c}}}$, i.e., the mapping between $(\mathcal{C}, \mathcal{S_{m^c}})$ and $\mathcal{X_{m^c}}$. We observe that a similar argument applies: Algorithm 2 dictates that the latent variables from the backtracking from $\mathcal{X_{m^c}}$ are either stored in either $\mathcal{C}$ or $\mathcal{S_{m^c}}$. It follows that $g_{\mathbf{x_{m^c}}}$ is invertible.

Condition 2: We show that $(\mathbf{c}, \mathbf{s_m}, \mathbf{s_{m^c}})$ returned by Algorithm 1 and Algorithm 2 satisfies Condition 2 by contradiction. We suppose that $\mathbf{s_m} \not\perp\!\!\!\perp (\mathbf{c}, \mathbf{s_{m^c}})$. Then it implied that $\exists \mathbf{d} \in (\mathbf{c}, \mathbf{s_{m^c}})$, $\exists \varepsilon \in \mathbf{s_m}$, such that $\mathbf{d} \in$ Descendants(ε). More precisely, it followed that there was a directed path that started from ε and ended at $\mathbf{d}$, and a child of ε, denoted as δ, was located on this path. If $\mathbf{d} \not\in$ Descendants(ε), there would be no directed paths from ε to $\mathbf{d}$ and thus at least one V-structure would sit on each path between ε

and $\mathbf{d}$ that blocked the path. According to Algorithm 1, as $\boldsymbol{\varepsilon} \in \mathbf{s_m}$, it implied that $\boldsymbol{\delta} \notin \mathbf{c}$ and $\boldsymbol{\delta} \notin \text{Ancestors}(\mathbf{x_m}) \cap \text{Ancestors}(\mathbf{x_{m^c}})$.

We first investigate the case where $\mathbf{d} \in \mathbf{c}$, i.e., $\mathbf{s_m} \not\perp\!\!\!\perp \mathbf{c}$. The fact that $\mathbf{d} \in \mathbf{c}$ implied that $\mathbf{d} \in \text{Ancestors}(\mathbf{x_m}) \cap \text{Ancestors}(\mathbf{x_{m^c}})$ which further implied that $\boldsymbol{\delta} \in \text{Ancestors}(\mathbf{x_m}) \cap \text{Ancestors}(\mathbf{x_{m^c}})$ as $\boldsymbol{\delta}$ was an ancestor of $\mathbf{d}$. Therefore, we have arrived at a contraction to the observation that $\boldsymbol{\delta} \in \text{Ancestors}(\mathbf{x_m}) \cap \text{Ancestors}(\mathbf{x_{m^c}})$.

We now discuss the scenario where $\mathbf{d} \in \mathbf{s_{m^c}}$. By design, Algorithm 2 ensures that $\mathbf{s_{m^c}}$ contains two types of latent variables, exogenous variables and a spouse of latent variables in $\mathbf{c}$. As $\mathbf{s_m}$ consists solely of exogenous variables and exogenous variables are independent mutually, it could only be the case that $\mathbf{d}$ was a spouse of a latent variable in $\mathbf{c}$. By Algorithm 1, there would be a directed path from $\boldsymbol{\delta}$ to $\mathbf{x_m}$. Also, Algorithm 2 ensured that $\mathbf{d}$ lied on a path directed to $\mathbf{x_{m^c}}$. As there existed a directed path from $\boldsymbol{\delta}$ to $\mathbf{d}$, there must exist a directed path from $\boldsymbol{\delta}$ to $\mathbf{x_{m^c}}$. Therefore, $\boldsymbol{\delta} \in \text{Ancestors}(\mathbf{x_m}) \cap \text{Ancestors}(\mathbf{x_{m^c}})$ which contradicts the fact established above.

Therefore, these contradiction implies that $\mathbf{s_m} \perp\!\!\!\perp (\mathbf{c}, \mathbf{s_{m^c}})$.

So far, we have shown that Algorithm 1 and Algorithm 2 yield $(\mathbf{c}, \mathbf{s_m}, \mathbf{s_{m^c}})$ that fulfills the conditions of Figure 3. In the following, we show that $(\mathbf{c}, \mathbf{s_m})$ is the minimal solution and is unique.

Uniqueness and minimality of $(\mathbf{c}, \mathbf{s_m})$: We now reason about that given the mask and the hierarchical structure, $(\mathbf{c}, \mathbf{s_m})$ returned by Algorithm 1 is the set of minimal dimensionality that can fulfill the conditions, and such a minimal set is unique.

By construction, Algorithm 1 ensures that for each $\mathbf{c} \in \mathcal{C}$ there exists an undirected path that is made up of a directed path from $\mathbf{c}$ to the masked variable $\mathbf{x_m}$ and a directed path from $\mathbf{c}$ to the unmasked variable $\mathbf{x_{m^c}}$ and no other $\mathbf{c}' \in \mathcal{C}$ sits on this entire undirected path. To see this, there must exist a directed path from $\mathbf{c}$ to $\mathbf{x_m}$ without any other $\mathbf{c}' \in \mathcal{C}$ on it, otherwise $\mathbf{c}$ would not be placed in $\mathcal{C}$ in Algorithm 1. In addition, the pruning stage of Algorithm 1 mandates that there must exist $\mathbf{x_{m^c}}$ such that the path from $\mathbf{c}$ to $\mathbf{x_{m^c}}$ does not contain other $\mathbf{c}' \in \mathcal{C}$. We note that $\mathbf{c}$ chosen by Algorithm 1 is the variable with the smallest possible dimension to block such a path, as it resides on the highest level compared to other variables on the path and the variable dimension increases monotonically along directed paths.

Therefore, the choice of each $\mathbf{c}$ is minimal, and such a choice is unique. As $\mathcal{S}_m$ is the set of exogenous variables necessary for $\mathcal{C}$ to restore $\mathcal{X}_m$, the selection of $\mathcal{S}_m$ is also unique. Hence, we conclude that the $(\mathcal{C}, \mathcal{S}_m)$ returned by Algorithm 1 is the minimal choice and is unique.

$\square$

B. Identifiability proof

In this section, we present the proof for Theorem 2. We first give a general identifiability theory (i.e., Theorem 3) for the generating process in Figure 3 and then make the connection to the proof of Theorem 2.

Theorem 3. *The generating process in Figure 3 is defined as follows:*

$$[\mathbf{v}_1, \mathbf{v}_2] = g(\mathbf{c}, \mathbf{s}_1, \mathbf{s}_2), \tag{3}$$

$$\mathbf{v}_1 = g_1(\mathbf{c}, \mathbf{s}_1), \tag{4}$$

$$\mathbf{v}_2 = g_2(\mathbf{c}, \mathbf{s}_2), \tag{5}$$

where $\mathbf{c} \in \mathcal{C} \subset \mathbb{R}^{d_c}$, $\mathbf{s}_1 \in \mathcal{S} \subset \mathbb{R}^{d_{s_1}}$, and $\mathbf{s}_2 \in \mathcal{S}_2 \subset \mathbb{R}^{d_{s_2}}$. Both g_1 and g_2 are smooth and have non-singular Jacobian matrices almost anywhere, and g is invertible.

If $\hat{g}_1 : \mathcal{Z} \to \mathcal{V}_1$ and $\hat{g}_2 : \mathcal{Z} \to \mathcal{V}_2$ assume the generating process of the true model (g_1, g_2) and match the joint distribution $p_{\mathbf{v}_1, \mathbf{v}_2}$, then there is a one-to-one mapping between the estimate $\hat{\mathbf{c}}$ and the ground truth $\mathbf{c}$ over $\mathcal{C} \times \mathcal{S} \times \mathcal{S}$, that is, $\mathbf{c}$ is block-identifiable.

Proof. For $(\mathbf{v}_1, \mathbf{v}_2) \sim p_{\mathbf{v}_1, \mathbf{v}_2}$, because of the matched joint distribution, we have the following relations between the true variables $(\mathbf{c}, \mathbf{s}_1, \mathbf{s}_2)$ and the estimated ones $(\hat{\mathbf{c}}, \hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2)$:

$$\mathbf{v}_1 = g_1(\mathbf{c}, \mathbf{s}_1) = \hat{g}_1(\hat{\mathbf{c}}, \hat{\mathbf{s}}_1), \tag{6}$$

$$\mathbf{v}_2 = g_2(\mathbf{c}, \mathbf{s}_2) = \hat{g}_2(\hat{\mathbf{c}}, \hat{\mathbf{s}}_2), \tag{7}$$

$$(\hat{\mathbf{c}}, \hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2) = \hat{g}^{-1}(\mathbf{v}_1, \mathbf{v}_2) = \hat{g}^{-1}(g(\mathbf{c}, \mathbf{s}_1, \mathbf{s}_2)) := h(\mathbf{c}, \mathbf{s}_1, \mathbf{s}_2), \tag{8}$$

where we define the smooth and invertible function $h := \hat{g}^{-1} \circ g$ that transforms the true variables $(\mathbf{c}, \mathbf{s}_1, \mathbf{s}_2)$ to estimates $(\hat{\mathbf{c}}, \hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2)$.

Plugging Equation 8 into Equation 6 yields the following:

$$g_1(\mathbf{c}, \mathbf{s}_1) = \hat{g}_1(h_{c,s_1}(\mathbf{c}, \mathbf{s}_1, \mathbf{s}_2)).$$

For $i \in \{1, \ldots, d_{v_1}\}$ and $(j \in \{1, \ldots, d_{s_2}\})$, taking partial derivative of the i-th dimension of both sides w.r.t. $s_{2,j}$:

$$0 = \frac{\partial g_{1,i}(\mathbf{c}, \mathbf{s}_1)}{\partial s_{2,j}} = \frac{\partial \hat{g}_{1,i}(h_{c,s_1}(\mathbf{c}, \mathbf{s}_1, \mathbf{s}_2))}{\partial s_{2,j}}.$$

The equation equals zero because there is no $s_{2,j}$ in the left-hand side of the equation. Expanding the derivative on the right-hand side gives:

$$\sum_{k \in \{1, \ldots, d_c + d_{s_1}\}} \frac{\partial \hat{g}_{1,i}}{\partial h_{(c,s_1),k}} \cdot \frac{\partial h_{(c,s1),k}}{\partial s_{2,j}}(\mathbf{c}, \mathbf{s}_1, \mathbf{s}_2) = 0 \tag{9}$$

For $(\hat{\mathbf{c}}, \hat{\mathbf{s}}_1) \in \mathcal{C} \times \mathcal{S} \setminus \mathcal{E}_1$ where $\mathcal{E}_1$ denotes some subset with zero measure, there are at least $d_c + d_{s_1}$ values of i for which vectors $[\frac{\partial \hat{g}_{1,i}}{\partial h_{(c,s_1),1}}(\hat{\mathbf{c}}, \hat{\mathbf{s}}_1), \ldots, \frac{\partial \hat{g}_{1,i}}{\partial h_{(c,s_1),d_c+d_{s_1}}}(\hat{\mathbf{c}}, \hat{\mathbf{s}}_1)]$ are linearly independent, which is equivalent to the non-singular Jacobian matrix condition. Therefore, the $(d_c + d_{s_1}) \times (d_c + d_{s_1})$ linear system is invertible and the solution states that:

$$\frac{\partial h_{(c,s_1),k}}{\partial s_{2,j}}(\mathbf{c}, \mathbf{s}_1, \mathbf{s}_2) = 0,$$

for any $k \in \{1, \ldots, d_c + d_{s_1}\}$, $j \in \{1, \ldots, d_{s_2}\}$, and $(\hat{\mathbf{c}}, \hat{\mathbf{s}}_1) \in \mathcal{C} \times \mathcal{S} \setminus \mathcal{E}_1$. Therefore, we have shown that h_{c,s_1}, i.e. $(\hat{\mathbf{c}}, \hat{\mathbf{s}}_1)$, does not depend on $\mathbf{s}_2$.

Applying the same reasoning to h_{c,s_2}, we can obtain that h_{c,s_2}, i.e. $(\hat{\mathbf{c}}, \hat{\mathbf{s}}_2)$ does not depend on $\mathbf{s}_1$ on $\mathcal{C} \times \mathcal{S}$.

Thus, for $(\hat{\mathbf{c}}, \hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2) \in \mathcal{C} \times \mathcal{S} \times \mathcal{S}$, we can observe that $\hat{\mathbf{c}}$ does not depend on $\mathbf{s}_1$ and $\mathbf{s}_2$, that is, $\hat{\mathbf{c}} = h_c(\mathbf{c})$.

Notice that in all procedures above, the roles of the true quantities $(\mathbf{c}, \mathbf{s}_1, \mathbf{s}_2, g, g_1, g_2)$ and the estimated quantities $(\hat{\mathbf{c}}, \hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \hat{g}, \hat{g}_1, \hat{g}_2)$ are symmetric. Therefore, we can switch the two sets of quantities and derive the relation: for $(\mathbf{c}, \mathbf{s}_1, \mathbf{s}_2) \in (\mathcal{C} \times \mathcal{S} \times \mathcal{S})$, $\mathbf{c}$ does not depend on $\hat{\mathbf{s}}_1$ and $\hat{\mathbf{s}}_2$, that is, $\mathbf{c} = h'_c(\hat{\mathbf{c}})$.

In sum, we have shown that on $(\mathcal{C} \times \mathcal{S} \times \mathcal{S})$, there is a one-to-one mapping between $\mathbf{c}$ and $\hat{\mathbf{c}}$.

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

We now show that Theorem 2 follows directly from Theorem 3.

Theorem 2. *(Identifiability of $\mathbf{c}$): For each mask $\mathbf{m}$, given the dimensions $(d_{\mathbf{c}}, d_{\mathbf{s_m}})$ the encoder function $E_{\mathbf{m}^c}(\cdot)$ recovers all information of $\mathbf{c}$ located in Theorem 1, i.e., there exists a one-to-one mapping h, s.t., $h(\mathbf{c}) = \hat{\mathbf{c}}$.*

Proof. We invoke Theorem 3 and establish the connection between the MAE training and the estimation model in Theorem 3. In particular, we show that under Assumption 2, any solution produced by the MAE objective satisfies the conditions in Theorem 3 and consequently is equipped with the identifiability guarantee.

We establish the correspondence between the MAE configuration and the estimation models in Theorem 3:

- $\mathbf{v}_1 \leftarrow \mathbf{x}_{\mathbf{m}}$;
- $\mathbf{v}_2 \leftarrow \mathbf{x}_{\mathbf{m}^c}$;
- $\hat{g}_1 \leftarrow D_{\mathbf{m}}(\cdot, \hat{\mathbf{s}}_{\mathbf{m}})$;
- $\hat{g}_2 \leftarrow \tilde{g}_{\mathbf{m}^c}$, where $E_{\mathbf{m}^c}(\cdot) = [\tilde{g}_{\mathbf{m}^c}^{-1}(\cdot)]_{1:d_c}$.

We can observe that the minimizer of MAE satisfies the conditions specified in Theorem 3. This is because for the optimal solution $E_{\mathbf{m}^c}$ of the MAE objective, we can always construct a $\tilde{g}_{\mathbf{m}^c}$, which, together with $D_{\mathbf{m}}$, matches the joint distribution $p_{\mathbf{x}_m, \mathbf{x}_{\mathbf{m}^c}}$ and shares $\hat{\mathbf{c}}$, as stipulated in Theorem 3. Thus, as shown in Theorem 3, there exists a one-to-one mapping between the MAE estimate $\hat{\mathbf{c}} := E_{\mathbf{m}^c}(\mathbf{x}_{\mathbf{m}^c})$ and the true variable $\mathbf{c}$, which concludes our proof. $\square$

C. Experimental Setup

In this section, we provide the details of the experimental setups for our empirical results. Checkpoints and some codes are in https://github.com/martinmamql/mae_understand.

C.1. Masked Autoencoder

Masked Autoencoder (MAE) is an auto-encoding approach based on Vision Transformers (ViT) [17]. It consists of five steps: masking, encoding, unmasking, decoding, and reconstruction. First, an image is divided into non-overlapping patches. Then MAE samples a subset of patches and discards the remaining patches. MAE uses a hyper-parameter, masking ratio, to determine the percentage of patches to discard. For instance, if the masking ratio is 75%, $\frac{3}{4}$ of the patches in an image will be discarded, and only $\frac{1}{4}$ of the patches will be fed into the encoder. The sampling of patches follows a uniform distribution. Next, a ViT encoder first embeds patches using a linear projection with

positional embeddings and then uses the processed embeddings to feed into transformer blocks. For decoding, MAE first re-arranges the encoded embeddings from the visible patches according to their corresponding positions in the original image and then uses a shared learned mask token to fill in the patches that are masked. Essentially, this means the input of the decoder is a combination of encoded visible patches and the mask tokens, where the positions of the mask tokens are the masked patches in the original image. The decoder is another lightweight ViT, and it processes the decoder input through transformer blocks. Lastly, the last layer of the decoder linearly projects output patches to pixels, and the pixel output is reshaped to form a reconstruction of the original image. The objective function is the mean squared error between the reconstruction and the original image. MAE has thrived because of its simple design and strong empirical performance.

In the main text, inspired by a follow-up work of MAE [28], we study the effect of masking by decoupling the patch size for masking images and the patch size hyperparameter in the ViT. Particularly, in the main text, we only vary the masking patch size and fix the ViT patch size at 8. Nevertheless, the original MAE [22] does not decouple the two patch sizes. Therefore, for the reference of readers, in Appendix, we provide some analysis and results produced based on the patch size design from the original MAE [22], where the masking patch size and the ViT patch size are equal. We study three patch sizes: $\{8, 16, 32\}$. The experimental setup in [28] and the setup in [22] are interchangeable except for whether the patch size for the Vision Transformer varies.

C.2. Pretraining and Linear Probing

For pretraining MAE under different masking ratios or patch sizes, we leverage the Tensor Processing Unit (TPU) from Google Cloud. We train separate MAE models for each (masking ratio, patch size) pair, and each pretrained MAE corresponds to a unique masking ratio and patch size. We train all MAEs for 800 epochs. Training time varies, with the shortest (patch size $= 32$) taking 18 hours on a TPU v3-128 Pod, and the longest (patch size $= 8$) taking 40 hours on a TPU v3-128 pod. The architecture follows the exact implementation from the original MAE paper [22], without any hyper-parameter tuning except masking ratio and patch size, which we study in this paper. Details of augmentation, initialization, and base learning rate scaling can be found in the Appendix section of [22], all of which we follow.

After pretraining, we also follow the original MAE work to use linear probing to evaluate the representation quality. After pretraining, we remove the projection layers and add a supervised learning classifier on frozen features of MAE encoders. The decoders are discarded during linear probing. Other details of linear probing can be found in the Appendix section of [22]. We use the same hyper-parameters of linear probing as in [22].

C.3. Reconstructing high-level or low-level representations

To perform reconstruction, we use both the encoder and the decoder from the pretrained MAEs. All samples from ImageNet-1K are passed through the encoder *without* any masking, and the decoder reconstructs images in the original input space. Since no masking is applied, no masking token is applied to the input of the decoder. We use the reconstructed images and the original images
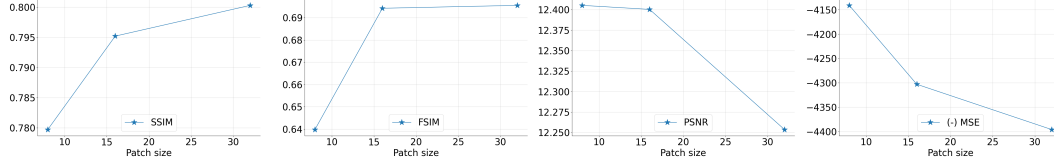
Figure 9. **Reconstruction evaluation** using the validation set without masking, based on two structural-level similarity metrics (SSIM and FSIM) and two pixel-level metrics (PSNR and MSE). We plot negative MSE for easier visualization. Higher SSIM and FSIM indicate high-level information is better captured, while higher PSNR and negative MSE indicates better low-level reconstruction. Here the patch size refers to the patch size in the original MAE, where the masking patch size and the patch size of ViT are equal.

to perform evaluations of four metrics: SSIM, FSIM, MSE, and PSNR. No training is performed, and the weights of the encoder and the decoder are frozen.

In Fig. 9, we show the reconstruction analysis using the original patch size design in MAE. Similar to the result in the main text, higher patch sizes produce image reconstructions capturing high-level similarities better, while low patch sizes have reconstructions better on low-level metrics.

C.4. Attention Analysis

We follow the attention heatmap visualization in DINO [10], where the chosen token is the [CLS] token or an object-related token. We visualize the self-attention module from the last block of the MAE encoder ViT. Brighter colors suggest larger attention weights. For easier visualization, attentions below a threshold of activation scores are not shown. We use the same threshold as [10]. For the self-attention visualization on the [CLS] token, we use an average of all heads in the last layer of the encoder ViT. For the self-attention visualization of the object-related token, we use the first head of the last layer of the encoder ViT, because using the average attention over all heads will result in a heatmap with much higher overall attention scores across pixels, making the visualization hard to interpret.
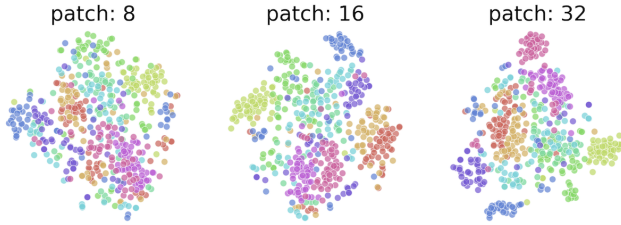

Figure 10. T-SNE embeddings of different MAE models under varied masking ratios and patch sizes. We fix the masking ratio at 0.75 to change patch sizes. Each color represents one ImageNet class. The patch size refers to the patch size in the original MAE, where the masking patch size and the patch size of ViT are equal.

C.5. Linear separability

To illustrate the linear separability of different MAEs under varied masking ratios or patch sizes, we sample ten random classes from ImageNet, and then use each MAE encoder to process images in the 10 classes to produce embeddings. We then project embeddings of all samples using PCA to a 50-dimension space

before t-SNE, as recommended by [56]. For t-SNE, we use a perplexity of 20.

In Fig. 10, we show the t-SNE plot using the original patch size design in MAE. Similar to the main text, embeddings are more separated in patch sizes 16 and 32 than 8, but differently, there are no significant differences between 16 and 32. Larger patch sizes generate more linearly separable embeddings in this case, although the separability seems indistinguishable for sizes 16 and 32.

For the robustness evaluation, we evaluate different variants of ImageNet validation datasets: ImageNet-v2 (INV2) [52], ImageNet-Adversarial (IN-A) [25], ImageNet-Rendition [4], and ImageNet-Sketch (IN-S) [59]. We also include another object classification dataset, ObjectNet (OJN) [4]. ImageNet-v2 contains three new test sets with 10,000 new images each, sampled a decade after the collection of the original ImageNet dataset, and is independent of existing models to prevent overfitting. ImageNet-Adversarial consists of natural images with adversarial filtration, meaning samples that can be classified with spurious cues are removed. Examples in ImageNet-A are harder to classify correctly and can cause mistakes across various models. ImageNet-Rendition contains renditions of ImageNet classes, such as art, cartoons, graffiti, and paintings. These examples share the same high-level object labels as ImageNet examples but differ in style and texture. ImageNet-Sketch contains black and white images of ImageNet classes, also differing in color and texture compared to original ImageNet samples. ObjectNet is a set of images captured at unusual poses in cluttered, natural scenes, which can severely degrade recognition performance.

Note that for evaluating these datasets, no training is performed; we use the MAE encoders *after* linear probings, therefore the checkpoints that are pretrained and linear-probed on ImageNet, and evaluate the checkpoints on these *validation* datasets without any parameter updates.

In Table 4, we show the robustness analysis using the original patch size design in MAE. A moderate patch size 16 yields the best robustness evaluation on IN-v2, OJN, IN-R, and IN-S. If we follow the original MAE and do not decouple masking patch size and ViT patch size, a medium patch size has stronger robustness performances than extreme patch sizes.

C.6. Shape bias

The cue-conflict dataset was introduced by [19] to evaluate how much deep learning models rely on shape information for prediction, which reflects the model's robustness to spurious correlation like textures. This dataset consists of 1280 images synthesized from 160 images of objects and 48 images of textures. The shape accuracy is measured by the fraction of images pre-

mask ratio	patch size	IN1K	IN-v2	OJN	IN-R	IN-A	IN-S
0.75	8	62.57	49.17	13.44	19.42	3.73	10.73
0.75	16	67.41	54.23	18.24	25.20	3.76	15.51
0.75	32	55.51	42.35	13.46	18.70	1.89	9.48

Table 4. **Accuracy (%) of linear probing and robustness evaluation** on ImageNet variants and ObjectNet. We linear probe MAE via supervised training on IN1K, and then perform inference on IN1K as well as other evaluation sets. We fix the masking ratio at 0.75 to change patch sizes. The patch size refers to the patch size in the original MAE, where the masking patch size and the patch size of ViT are equal.

mask ratio	patch size	AP^{box}	AP^{mask}
0.75	8	34.21	32.28
0.75	16	33.77	32.04
0.75	32	32.39	30.54

Table 5. **COCO object detection and segmentation** using a ViT Mask R-CNN baseline. We fix the masking ratio at 0.75 to change patch sizes. The patch size refers to the patch size in the original MAE, where the masking patch size and the patch size of ViT are equal.

dicted correctly by their shape. We directly run the pretrained MAE models with linear probes trained on ImageNet-1K on the cue-conflict dataset to examine the representation resulting from MAE pretraining without any adaptation to the test dataset.

C.7. Transfer learning

We use the pretrained MAE ViT encoder as an FPN [42] backbone in Mask-RCNN [24], following [22]. To do so, [22] uses a stack of pretrained transformer blocks in MAE to produce feature maps at a single scale; for instance, patch size 16 will produce stride 16 features. Then the features are equally divided, and upsampling or downsampling is applied to create features at different scales. Lastly, the FPN is built on multi-scale features. Below we include the transfer learning results of different patch sizes on COCO object detection and segmentation [43]. Because different patch sizes in ViT will influence the scale of feature maps in the FPN, we enforce the same combinations of multi-scale features: i.e., stride 4, 8, 16, and 32.

From Table 5, we show the transfer learning results of MAE under different patch sizes. Patch size 8 performs the best, and patch size 16 is better than 32. The reason for the better performance at patch size 8 may be due to a smaller batch size used, compared to patch size 16 and 32 (we can only fit batch size 1 for patch size 8 due to the increased number of tokens to process because of a smaller patch size.) We use the same batch size for 32 and 16, and the comparison between the two supports our claim: an extreme masking scheme can hurt the model's capacity to capture high-level information or, in this case, the semantic understanding of the scene.