Tutorial on Joint Embedding Predictive Architectures (JEPA): Foundations, Applications, and Future Directions

MEHDI MONEMI, Centre for Wireless Communications, University of Oulu, Finland
MARYAM CHINIPARDAZ, Jundi-Shapur University of Technology, Dezful, Iran
MEHDI RASTI, Centre for Wireless Communications, University of Oulu, Finland
MEHDI BENNIS, Centre for Wireless Communications, University of Oulu, Finland
MATTI LATVA-AHO, Centre for Wireless Communications, University of Oulu, Finland

Joint-Embedding Predictive Architectures (JEPAs) have recently emerged as a unifying paradigm in self-supervised representation learning, combining the semantic alignment of joint-embedding methods with predictive modeling in latent space. This tutorial provides a comprehensive and systematic exposition of JEPA and its extensions, covering its theoretical foundations, architectural design principles, and diverse application domains. We first situate JEPA within the broader taxonomy of representation learning and formulate its core components, including context-target generation, encoding, latent-space prediction, regularization, and energy minimization. Various JEPA applications are also elaborated ranging from downstream tasks facilitated by JEPA to system 2 planning and decision-making via predictive world models. A comprehensive survey of existing JEPA implementations in the literature across various modalities including image, audio, video, point-cloud, and multimodal applications is also presented. The tutorial also surveys emerging domain-specific applications of JEPA in 6G networks, where only a few pioneering studies exist to date. Finally, open challenges and research directions for advancing JEPA in various domains are discussed.

CCS Concepts: • **Computing methodologies → Learning latent representations**.

Additional Key Words and Phrases: Joint Embedding Predictive Architecture, JEPA, representation learning, world models, agentic AI, semantic communication, image processing, wireless networks

ACM Reference Format:
Mehdi Monemi, Maryam Chinipardaz, Mehdi Rasti, Mehdi Bennis, and Matti Latva-aho. 2026. Tutorial on Joint Embedding Predictive Architectures (JEPA): Foundations, Applications, and Future Directions. *ACM Comput. Surv.* 1, 1 (June 2026), 38 pages. https://doi.org/10.1145/XXXXXXX.XXXXXXX

1 Introduction

Representation learning broadly refers to the process of learning a transformation from the original data domain (e.g., pixels, audio samples, or sensor readings) into another domain, typically a continuous latent space referred to as *representations* or *embeddings*[1]. The learned embeddings may capture different levels of abstraction depending on the learning objective. Of particular interest are those that encode high-level, task-relevant abstractions that remain consistent across variations in viewpoint, scale, or modality, referred to as *semantic embeddings*. Such embeddings provide a compact and meaningful description of the underlying data [5, 11, 56].

As illustrated in Fig. 1, (label-free) representation-learning methods can be organized hierarchically along three conceptual dimensions. At the first level lies the distinction between *unsupervised*

[1]While in some contexts the terms *representation* and *embedding* carry slightly different technical meanings, we use them interchangeably throughout this tutorial.

Authors' Contact Information: Mehdi Monemi, mehdi.monemi@oulu.fi, Centre for Wireless Communications, University of Oulu, Oulu, Finland; Maryam Chinipardaz, m.chinipardaz@jsu.ac.ir, Jundi-Shapur University of Technology, Dezful, Iran; Mehdi Rasti, mehdi.rasti@oulu.fi, Centre for Wireless Communications, University of Oulu, Oulu, Finland; Mehdi Bennis, mehdi.bennis@oulu.fi, Centre for Wireless Communications, University of Oulu, Oulu, Finland; Matti Latva-aho, matti.latva-aho@oulu.fi, Centre for Wireless Communications, University of Oulu, Oulu, Finland.

2026. ACM 1557-7341/2026/6-ART
https://doi.org/10.1145/XXXXXXX.XXXXXXX

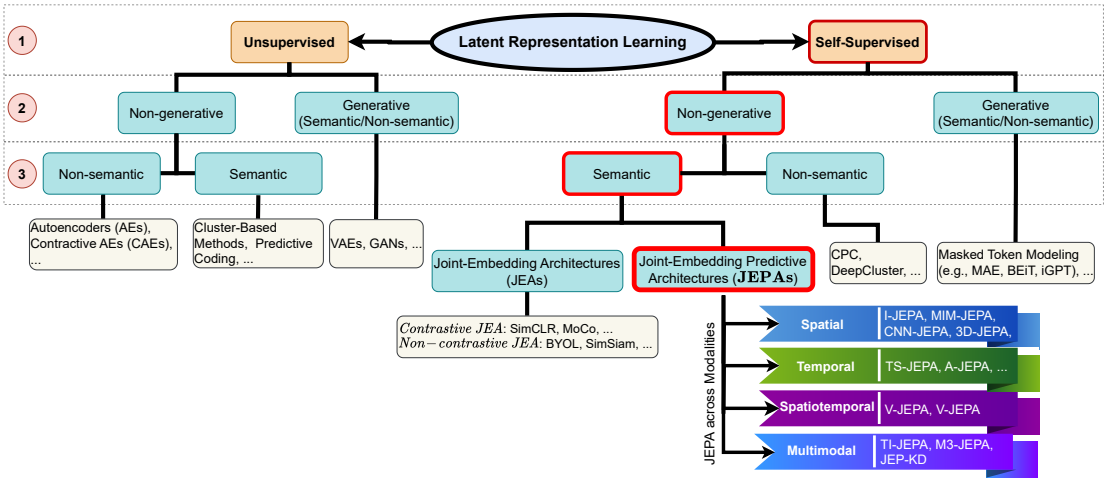

Fig. 1. Latent representation learning architectures including JEPA

and *self-supervised* learning. *Unsupervised learning* such as Autoencoders (AEs) [47] and Variational Autoencoders (VAEs) [55] methods aim to extract structure from unlabeled data without relying on external supervision signals. In contrast, self-supervised learning (SSL) defines *pretext tasks* or *surrogate objectives* from the data itself to enable the model to learn useful representations from unlabeled data. Typical examples include reconstructing masked or corrupted regions in colorization, inpainting, or Masked Autoencoders (MAE) [44, 73, 99], or aligning augmented views of same inputs as in SimCLR and MoCo [19, 45]. Fig. 2a–b illustrates the difference between conventional (unsupervised) autoencoding (e.g., AE or VAE) and self-supervised masked autoencoding (MAE). In an AE/VAE, the model learns a latent representation by reconstructing the input, typically through a loss of the form $\mathcal{L} = D(\hat{I}, I)$ where D is a distance metric and $\hat{I}$ is the decoded reconstruction of input I. In contrast, MAE as a representative SSL scheme defines a surrogate prediction task: the input is divided into visible context x and masked target y, and the model is trained to predict the missing target from the context, leveraging $\mathcal{L} = D(\hat{y}, y)$. When needed, auxiliary information z such as the positions of masked patches can condition this prediction to enable more accurate results. This shift from full-input reconstruction to context-target prediction is an important step toward predictive self-supervised representation learning.

The second level in Fig. 1 distinguishes between *generative* and *non-generative* representation-learning paradigms. *Generative methods* such as VAEs [55], and Generative Adversarial Networks (GANs) [36] learn by reconstructing or synthesizing data samples in the *original input domain*. Their training objectives typically minimize a reconstruction or adversarial loss between the input and its regenerated version, requiring high-capacity decoders to preserve pixel- or sample-level fidelity. In contrast, *non-generative methods* discard the need for explicit reconstruction and instead learn directly within the *representation* space. At the third level, methods can further be differentiated as *semantic* or *non-semantic*. Early unsupervised models typically encoded low-level statistical regularities, whereas many recent self-supervised approaches aim to produce *semantic embeddings*, as representations that capture invariant, task-relevant structure and generalize across transformations, modalities, or domains [5, 11, 56]. Consequently, much recent research has focused on **semantic, non-generative, self-supervised** architectures, which combine the abstraction strength of semantic learning with the efficiency of non-generative training [19, 44, 56].

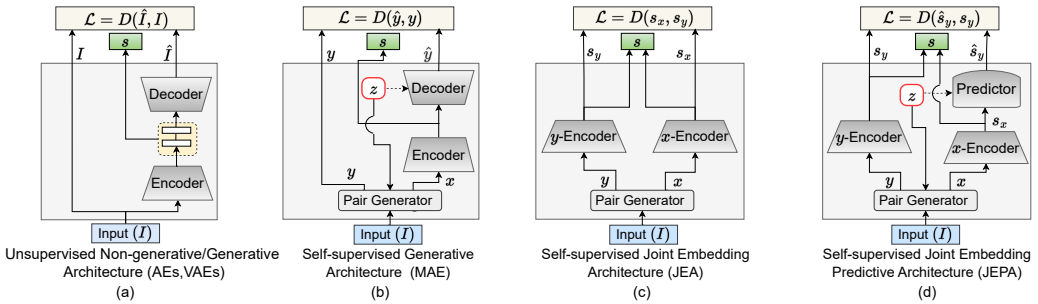

Fig. 2. Comparison of JEPA (d) with other representation learning architectures (a-c)

Within this region, a major milestone was achieved with the development of **Joint Embedding Architectures (JEAs)**. As illustrated in Fig. 2c, JEAs generate a pair (x, y) from the original input I, where each element of the pair is processed by a separate encoder, and the resulting embeddings are aligned within a shared latent space using a loss function $D(s_x, s_y)$, where s_x and s_y are the representations of x and y respectively obtained from the corresponding encoders. JEAs can be *contrastive* or *non-contrastive*. *Contrastive* JEAs (e.g., SimCLR and MoCo in [19, 45]) enforce this alignment by distinguishing between *positive pairs* (e.g., different augmentations of the same input) and *negative pairs* (e.g., augmentations of different inputs). *Non-contrastive* variants (e.g., BYOL and SimSiam in [20, 38]) achieve similar alignment objectives (without relying on explicit negative samples) by preventing *representational collapse* through schemes discussed later in this article. Despite this progress, conventional JEAs remain *alignment-based* rather than *predictive*: they learn to match related representations but do not explicitly model the transformation or dependency between them. To address this limitation, LeCun [56] formalized the **Joint Embedding Predictive Architecture (JEPA)**, an idea whose roots trace back to Schmidhuber and Prelinger [79], who proposed *Predictability Maximization* (PMAX) in 1993 for training one network to predict another network's latent representation rather than its raw input. JEPA augments the JEA framework with a *predictor* that learns to estimate the latent representation of a target view y from that of a context view x. The prediction occurs entirely within the latent embedding space, making JEPA a **non-generative** architecture that learns semantic relationships and predictive abstractions without reconstructing the original input. As visualized in Fig. 2, JEA (Fig. 2c) aligns the embeddings of x and y leveraging the loss function $D(s_x, s_y)$. JEPA (Fig. 2d) changes this architecture by predicting the embeddings of y (denoted by $\hat{s}_y$) via minimizing the error $D(\hat{s}_y, s_y)$, using the embeddings of x (i.e., s_x). This enables modeling higher-level semantic and data structural dependencies, fully elaborated in this work.

The remainder of this paper provides a comprehensive tutorial on JEPA, its architecture and extensions, various applications in different domains and modalities, along with a literature review of existing frameworks. The paper roadmap is visualized in Fig. 3.

2 JEPA Framework and Methodology

JEPA encourages the model to capture high-level semantic structure without relying on direct input reconstruction. In a nutshell, JEPA couples two core objectives:

- **Self-supervised latent predictability:** The representation of a perturbed or transformed view of an input (the *target*) should be predictable in the latent space leveraging the representation of the original or another transformed view (the *context*) of the same sample.

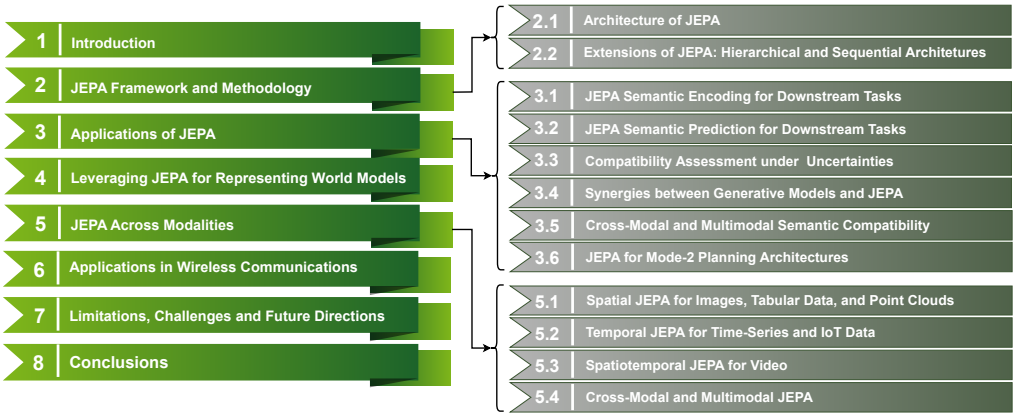

Fig. 3. Roadmap of the tutorial: top-level sections and representative subsections.

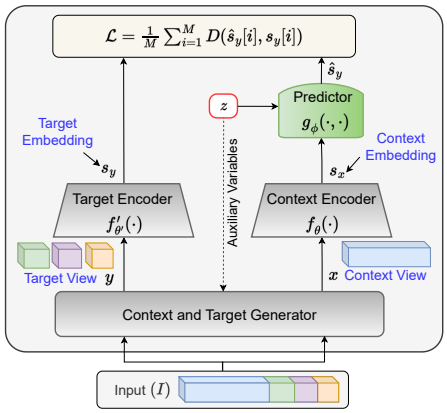

Fig. 4. JEPA architecture in the training phase.

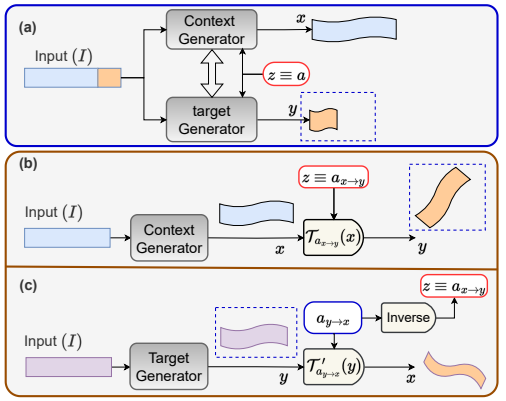

Fig. 5. View generation (a) *split-view* (b) *transformation based* where context is first generated. (c) *transformation based* where target is first generated.

- **Anti-collapse constraint:** Learning the prediction of targets' representations must avoid degenerate and trivial solutions; while predicting targets' representations, sufficient variations in the representation space should be maintained for meaningfully different inputs, so that semantically distinct inputs remain distinguishable in the representation space.

Fig. 4 illustrates the generic structure and key components of the JEPA framework, highlighting the flow diagram that governs the training phase of JEPA. Based on this structure, the main steps in the JEPA training pipeline are detailed in the following subsections. To provide the reader with a conceptual representation of the concepts as well as some practical implementation techniques, we also refer, where appropriate, to I-JEPA, a representative implementation, which demonstrates the effectiveness of JEPA for learning semantic representations in image datasets [5].

2.1 Architecture of JEPA

The typical structure of a JEPA model is illustrated in Fig. 4, and its components are discussed in the following subsections.

2.1.1 Context and Target Views Preparation. Consider a dataset of semantically rich data samples. In line with other self-supervised learning (SSL) methods, during the training phase, JEPA decomposes each input sample I into a set of $M \geq 1$ pairs, denoted by $\{(x[i], y[i])\}_{i=1}^{M}$. We refer to $x[i]$ as the *context view* and $y[i]$ as the *target view* corresponding to the i-th pair. For notational simplicity, we will often use (x, y) to represent any context-target pair $(x[i], y[i])$ associated with the input I. The context-target pairs (x, y) form the basis of the predictive learning task, in which the model estimates the latent representation of a target view y from that of the corresponding context view x within a shared embedding space, as will be elaborated in the next sections. The views are either available in advance, or generated by an internal pair-generator module illustrated in Fig. 4, through techniques such as masking (e.g., hiding portions of the input). Typically, the context and target views correspond to disjoint parts of I (e.g., separate groups of pixels in an image or separate segments in a sequence). Since x and y originate from the same input, they are inherently correlated; however, ensuring that this correlation is preserved at a meaningful level requires careful design of the view-generation strategy for both context and target views.

In practice, the input sample I may be a single entity (e.g., an image) or a sequence of related entities (e.g., a video $I = \{I_1, I_2, \ldots, I_N\}$ consisting of N frames). Two main paradigms of view generation can be identified, as conceptually illustrated in Fig. 5.

(1) **Split-view generation**, adopted in models such as I-JEPA [5] and many of its precedent versions, where both the context and target views are derived directly by dividing the input I into non-overlapping subregions and assigning some subregion to the context and target views. As illustrated in Fig. 5-a, each view is obtained by selecting distinct subsets or segments of the original input and potentially applying stochastic augmentations in the data domain (for example, pixel- or patch-level masking, cropping, or color jitter for images, and temporal masking for audio or sequential signals). In this scheme, the structural relation between the two views is *implicit* and is also reflected through auxiliary metadata z that conveys their relative geometric, positional, or temporal correspondence. As a simple example, consider an input vector I in which certain elements are randomly masked to form x, while the elements corresponding to the masked indices correspond to y. In this case, the auxiliary variable z can correspond to the positional indices of the masked elements.

(2) **Transformation-based generation**, employed in perceptual world-model formulations such as the Image World Model (IWM) [30], where one view is obtained directly from the original input sample I (either by taking the whole sample or an augmented version of it), and the second view is explicitly generated by applying a transformation $\mathcal{T}_a$. The variable a denotes the parameters of the transformation (e.g., cropping window, rotation angle, or color jitter strength) and serves as the auxiliary conditioning variable ($z \equiv a$) used to guide the predictor. In general, as illustrated in Figs. 5b and 5c, the transformation between the two views can be expressed in either direction:

$$y = \mathcal{T}_{a_{x\to y}}(x), \qquad x = \mathcal{T}'_{a_{y\to x}}(y), \tag{1}$$

where $\mathcal{T}_{a_{x\to y}}$ and $\mathcal{T}'_{a_{y\to x}}$ denote the context-to-target and target-to-context transformations parameterized by $a_{x\to y}$ and $a_{y\to x}$, respectively. These correspond to the cases where the context view x or the target view y is generated first.

To illustrate the split-view generation scheme in JEPA, we describe the image-based I-JEPA framework [5]. From each input image I, I-JEPA constructs a single context view x and $M = 4$ target views $y[i]$, yielding context-target pairs $\{(x, y[i])\}_{i=1}^{4}$ (Fig. 6). The four target views are rectangular blocks with scale uniformly sampled from $(0.15, 0.2)$ and aspect ratio from $(0.75, 1.5)$, each covering 15-20% of the image area. This variability produces diverse horizontal and vertical

Fig. 6. I-JEPA context and target views for two sample images

regions, promotes structural learning, and prevents overfitting and trivial solutions. The context view is then sampled independently as a single rectangular region with scale $(0.85, 1.0)$ and fixed aspect ratio 1.0. Since context and target regions are sampled independently, they may overlap; overlapping areas are excluded from the final context view to ensure a meaningful prediction task. For the case of image-based JEPA, the view-generation strategies can be categorized within the broader family of masking approaches developed in the masked image modeling (MIM) literature. In these methods, an image is first partitioned into a grid of non-overlapping patches, and masking is then applied at the patch level. Two of the most common approaches are *random masking*, where patches are sampled uniformly until a given masking ratio is reached, as in MAE [44] and SimMIM [95], and *block-wise masking*, where masking is applied to contiguous blocks of patches (often rectangular in shape), as in BEiT [9] and I-JEPA [5]. I-JEPA adopts a multi-block masking scheme, in which a 224×224 image is split into $14 \times 14 = 196$ non-overlapping patches of size 16×16, and the context and target views are then obtained by sampling blocks of these patches according to specified scale and aspect ratio constraints, where the patches corresponding to the target blocks are masked.

In summary, the design of context-target generation in JEPA should ensure that targets are semantically rich and disjoint, while the context provides sufficiently correlated information to enable accurate prediction of the target embeddings.

2.1.2 Context and Target Encoding. After view generation, JEPA encodes the context view x and each target view y into latent embeddings via separate encoders:

$$s_x = f_\theta(x), \quad s_y = f'_{\theta'}(y). \tag{2}$$

The context encoder f_θ and target encoder $f'_{\theta'}$ may differ in architecture enabling cross-modal applications (e.g., JEP-KD [85] aligns video context with audio targets). However, in same-modality settings, the encoders are typically identical in design, but θ' is updated as an exponential moving average (EMA) of θ:

$$\theta' \leftarrow \tau\theta' + (1-\tau)\theta, \tag{3}$$

with momentum $\tau \in [0, 1)$. In I-JEPA, τ starts at 0.996 and linearly increases to 1.0 [4, 16]. This EMA implicitly applies stop-gradient for target-encoder aiming to stabilize training and preventing representation collapse as detailed in Section 2.1.4).

2.1.3 Predictor Design and Loss Function. Following the encoding of context and target views into latent representations $s_x = f_\theta(x)$ and $s_y = f'_{\theta'}(y)$, a predictor network g_ϕ estimates the target representation as

$$\hat{s}_y = g_\phi(s_x; z), \tag{4}$$

where z denotes auxiliary variables to improve the prediction. We can interpret (4) as effectively performing *semantic embedding in-painting*, meaning that if the target embedding s_y is missed or

masked within the representation $s = (s_x, s_y)$, it can be predicted and in-painted to reconstruct the whole representation $(s_x, \hat{s}_y)$.

In general, the auxiliary variables z fall into two categories:

- *Conditioning variables*: These are sampled randomly during view generation as $z \sim \mathcal{P}_z$, but when realized, they are treated as known to the predictor. They function as conditioning tokens that guide the prediction task. For example, in I-JEPA, the positional information of the masked patches form the auxiliary variables to help in predicting the target embedding.
- *Unobservable stochastic variables*: These are latent random variables that cannot be observed directly by the predictor but represent latent uncertain factors influencing the prediction process. Leveraging energy-based models [6, 57], their role is taken into account by minimizing an energy metric over the domain of such variables in the latent space, effectively selecting the most compatible latent configuration.

Since unobservable stochastic latents primarily discussed in [56] are rarely employed in current JEPA implementations, we defer their discussion to Section 3.3. Hereafter, the term *auxiliary variables* will, unless otherwise specified, refer to *conditioning variables*. Note, however, that such auxiliary variables are not required in all JEPA implementations. In some designs, the context-target pairs are already available and/or sufficiently informative so that the target can be predicted from the context *without any explicit auxiliary variables*. Unlike contrastive SSL methods, which rely on distinguishing positive from negative pairs, a process which is often impractical in many applications, JEPA adopts a non-contrastive approach that directly predicts the target representation from the context in latent space, eliminating the need for negative samples. This is leveraged through assessing context-target compatibility via an Energy-Based Model (EBM), where a scalar energy function $E(x, y)$ (or more generally $E(x, y, z)$ when auxiliary information z is used) assigns low energy to compatible pairs and high energy to incompatible ones.

In practice, however, the high dimensionality of x and y makes direct evaluation in the original input space inefficient, motivating the formulation of the energy in the representation space as $E(s_x, s_y, z)$. Accordingly, the JEPA energy for a context-target compatibility measure from the representation pair (s_x, s_y) conditioned on auxiliary variable z is defined as the target-prediction error in the latent space in terms of a distance metric $D(\cdot, \cdot)$, as follows:

$$E(s_x, s_y; z) = D(\hat{s}_y, s_y) = D(g_\phi(s_x; z), s_y). \tag{5}$$

Example (Masked Image Prediction): To interpret (5), consider a toy image I divided into four patches $\{p_1, p_2, p_3, p_4\}$. Suppose p_4 is the target (masked) region and $\{p_1, p_2, p_3\}$ form the context. The context encoder maps the three visible patches to $s_x = f_\theta(\{p_1, p_2, p_3\})$, while the target encoder produces $s_y = s_4 = f'_{\theta'}(p_4)$. The auxiliary variable z, which in practice is a token embedding of the positional index 4 (written here as $z{=}4$ for brevity), is incorporated into the predictor to enable a more accurate prediction. The predictor then computes $\hat{s}_4 = g_\phi(s_x;\ z{=}4)$, and the energy in (5) can be written as $E(s_x, s_4, 4) = \|\hat{s}_4 - s_4\|_2^2$, which also forms the loss function to be minimized during training, noting that s_4 is the naturally compatible target here for the given context. Lower energy thus indicates that the predicted embedding $\hat{s}_4$ is more compatible with the true target embedding s_4. Moreover, since the predictor here is trained on patches from the same input I, it learns a representation specific to its structure; consequently, replacing s_4 with $\tilde{s}_4 = f'_{\theta'}(\tilde{p}_4)$ corresponding to a patch $\tilde{p}_4$ from a different image $\tilde{I}$, is expected to yield a higher energy $E(s_x, \tilde{s}_4, 4) > E(s_x, s_4, 4)$, reflecting the potential incompatibility between predicted embedding and foreign target embedding.

As pointed out, the predictor together with the encoders are trained to minimize the energy defined in (5). In general however, we may have more than one target corresponding to each input.

Given an input I comprising M context-target view pairs, and adopting the squared ℓ_2-norm as the distance metric (as is common practice in most JEPA works), the training loss can be written as

$$\mathcal{L} = \frac{1}{M}\sum_{i=1}^{M} D(\hat{s}_y[i], s_y[i]) = \frac{1}{M}\sum_{i=1}^{M} \|g_\phi(s_x[i]; z[i]) - s_y[i]\|_2^2 \tag{6}$$

In general, the auxiliary variables z can be tokenized and incorporated into the predictor network in two ways: either by concatenating their embeddings with the context embeddings s_x at the feature level, or by prepending them as separate tokens to the input sequence of a transformer-based predictor. The latter approach, adopted in large models such as I-JEPA, allows the predictor to integrate conditioning information through self-attention applied over the full token sequence. As an illustrative example of auxiliary variables, for each input sample image, I-JEPA generates $M = 4$ context-target pairs $(x, y[i])$, $i \in \{1, 2, 3, 4\}$, each associated with auxiliary tokens $z[i]$. These tokens explicitly encode positional and information of masked tokens, which enables the predictor to implicitly infer additional information relating to scale and aspect ratios of the views. While the masked patches are sampled randomly, the observed realizations relating the positional information of masked patches are embedded and fed to the predictor. A more detailed account of how these embeddings are constructed and integrated in I-JEPA will be given in Section 5.1.

More generally, noting that for each sample I various contexts and targets can be sampled according to some random view generation policy, (per sample) loss function can be represented as

$$\mathcal{L} = \mathbb{E}_{(x,y)\sim\mathcal{P}_{(x,y)}}\left[D(\hat{s}_y, s_y)\right] \equiv \mathbb{E}_{z\sim\mathcal{P}_z}\left[D(g_\phi(s_x; z), s_y)\right], \tag{7}$$

where the parameters optimized by backpropagation are the predictor parameters ϕ and the context-encoder parameters θ, while the target-encoder parameters θ' are typically updated via techniques such as EMA to prevent representational collapse. The expectation is over the stochastic view-generation process. Specifically, $\mathcal{P}_{(x,y)}$ denotes the joint distribution of context–target pairs induced by the sampling policy for each input I; it is not usually learned by backpropagation, but chosen as a design or hyperparameterized policy controlling the diversity and difficulty of the prediction task. In practice, the expectation is approximated by averaging over a finite number of sampled pairs for each input, e.g., $M = 4$ target views per image in I-JEPA according to (6), where the context-target sampling policy governs the random locations of target/context blocks and their scale and aspect ratios.

2.1.4 Anti-Collapse Regularization in JEPA. A purely predictive JEPA objective can admit degenerate solutions where many of the inputs are mapped to very similar latent vector, yielding perfect prediction but useless representations. Anti-collapse regularization is therefore crucial: it constrains the embeddings so that semantically different inputs remain distinguishable while the predictor still learns to match target latents. Letting (x_n, y_n) be a sampled context-target pair relating to the input sample n of a batch of N samples, the objective is to minimize the overall loss function represented as follows:

$$\mathcal{L}^{\text{tot}} = \sum_{n=1}^{N} \mathbb{E}_{(x_n,y_n)\sim\mathcal{P}_{(x_n,y_n)}}\left[D(\hat{s}_{y_n}, s_{y_n})\right] + \lambda \times \mathcal{R}\left((s_{x_n}, s_{y_n})_{n\in[N]}\right) \tag{8}$$

where the parameters optimized by backpropagation are the predictor parameters ϕ and the context-encoder parameters θ, noting that the target-encoder parameters θ' may be learned directly or updated separately, e.g., by EMA. Here, the first term represents the latent predictive invariance of JEPA, the second term accounts for characterizing the anti-collapse regularization, and λ is the regularization hyperparameter that balances prediction accuracy against embedding diversity. In this regard, there are two important parameters affecting the regularization, (a) **sampling**

distributions $\mathcal{P}_{(x_n, y_n)}, n \in [N]$, which controls the paired embeddings (s_{x_n}, s_{y_n}) fed into the regularization evaluator $\mathcal{R}(\cdot)$ and (b) **formulation of** $\mathcal{R}(\cdot)$, which quantifies how well given sample pairs are regularized.

Broadly, JEPA models prevent collapse using one or more of the following schemes:

• *Teacher-Student Schemes:* This paradigm employs asymmetric architectures where a teacher network (kept under stop-gradient and typically updated as a slow EMA of the student) provides stable targets for prediction, implicitly promoting diversity by decoupling the update dynamics and preventing trivial constant solutions. Many non-JEPA (e.g., BYOL [38] and DINO [16, 71]) and JEPA (e.g., I-JEPA, CNN-JEPA, V-JEPA) schemes leverage this technique.

• *Non-Parametric Estimators:* These methods (e.g., SimCLR [19], MoCo [45], and CLIP [77]) use contrastive, non-parametric objectives that pull together embeddings of positive pairs (e.g., different augmented views of the same input) while pushing apart embeddings of negative pairs (e.g., views from different inputs in the batch).

• *Moment-Matching Objectives:* These methods explicitly regularize the embeddings by enforcing low-order batch statistics (e.g., variance and covariance) to form well-behaved target distribution (typically isotropic Gaussian). Prominent examples include VICReg [12] and W-MSE [24]. For example, C-JEPA [64] augments the JEPA loss (6) with VICReg's three terms: *invariance* (aligning embeddings of different augmented views of the same input), *variance* (preventing any embedding dimension from becoming nearly constant across samples in a batch), and *covariance* (decorrelating features by minimizing off-diagonal entries of the covariance matrix). A recent development in this direction is the introduction of Sketched Isotropic Gaussian Regularization (SIGReg) in LeJEPA [8]. Similar to VICReg, SIGReg controls embedding statistics to prevent collapse, but instead of separately enforcing variance and covariance constraints, it regularizes embeddings toward an isotropic Gaussian distribution. Compared with VICReg, SIGReg provides a more global distribution-matching regularizer, which also yields a JEPA objective with a single trade-off hyperparameter, and remain stable across parameter settings, architectures, and domains. This mechanism is later used in recent JEPA-based world model work LeWM [62].

2.2 Extensions of JEPA: Hierarchical and Sequential Architectures

The original JEPA framework can be extended to tackle real-world complexities through multi-layer hierarchical and sequential paradigms. These extensions enhance the model's capability to handle multi-resolution abstractions and temporal dynamics.

The Hierarchical JEPA (H-JEPA), depicted in Fig. 7, offers a multi-level extension where interconnected JEPA modules operate across varying scales of abstraction, enabling multi-resolution reasoning and abstraction presentations. Each level n encodes context embeddings $s_x[n-1]$ obtained from the level $n-1$ in to a higher level of abstraction $s_x[n]$ and predicts corresponding target representation $\hat{s}_y[n]$ using a dedicated predictor $g_\phi^{(n)}$, with training conducted level-wise or globally across levels [56]. This hierarchical structure enables the model to progressively abstract fine-scale details at higher levels, producing generalized latent representations that facilitate complex, long-horizon tasks. For instance, a two-level H-JEPA model might extract detailed latent features at the lower level optimized for short-term predictions, while a higher level generates coarser, more abstract embeddings to support longer-term forecasting and planning.

In contrast to H-JEPA, the Sequential JEPA (S-JEPA), illustrated in Fig. 8, arranges a chain of predictors, where each module n generates the next step target representation prediction $\hat{s}_y[n+1]$ based on the current prediction $\hat{s}_y[n]$ and additional latent auxiliary variables $z[n]$. This iterative process enables step-by-step simulation of future representations, drawing inspiration from model

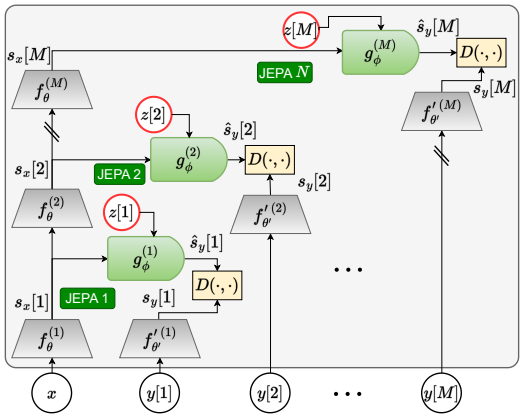

Fig. 7. Hierarchical JEPA (H-JEPA) with multiple JEPA modules at different representation levels.

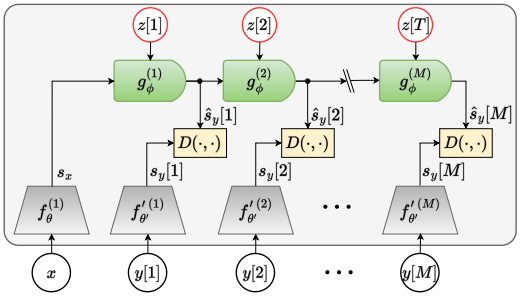

Fig. 8. Sequential JEPA (S-JEPA) employing a chain of successive predictor modules.

predictive control [63] and model-based reinforcement learning [66], which evaluate action sequences through recursive state updates. This concept parallels the approach used in MuZero, which employs specialized networks dedicated to modeling dynamics, rewards, and policies, enabling effective planning and decision-making in complex environments [80]. This architecture is particularly well-suited for applications requiring forecasting and planning at abstract representation levels over successive time-step windows, including robotic navigation, video frame prediction, and other sequential decision-making and planning tasks.

Besides the simplistic S-JEPA and H-JEPA models where a single context is paired to a sequence of targets, we can design more advanced settings where multiple context views and multiple target views are coupled to capture more complex representation alignment configurations.

3 Applications of JEPA

Fig. 9 summarizes a set of representative application categories for JEPA. In the following, we elaborate on each item.

3.1 JEPA Semantic Encoding for Downstream Tasks

A key strength of JEPA is its ability to produce semantically rich embeddings that transfer effectively to *downstream tasks*, including classification, object detection, and density estimation. We first demonstrate its application to the canonical task of classification, then discuss how it can support more unconventional downstream task such as density estimation.

3.1.1 JEPA for Classification. Fig. 9A illustrates three paradigms for supervised classification. In the simplest structure (a), classification is performed directly on raw high-dimensional data domain, without any form of dimensionality reduction. A second approach (b) incorporates dimensionality reduction techniques to first obtain compact embeddings, which are then fed to a classifier. Representative examples of *unsupervised* compressors typically leveraged for classification include principal component analysis (PCA) [51] as a classical linear technique, t-SNE [89] as a nonlinear manifold-preserving alternative, and autoencoders [47] as neural-network-based compressors.

JEPA-based approach (c) offers a framework that provides a self-supervised semantic and compact representation learning for downstream tasks. Typical training pipeline for JEPA-based classification schemes follows the following steps:

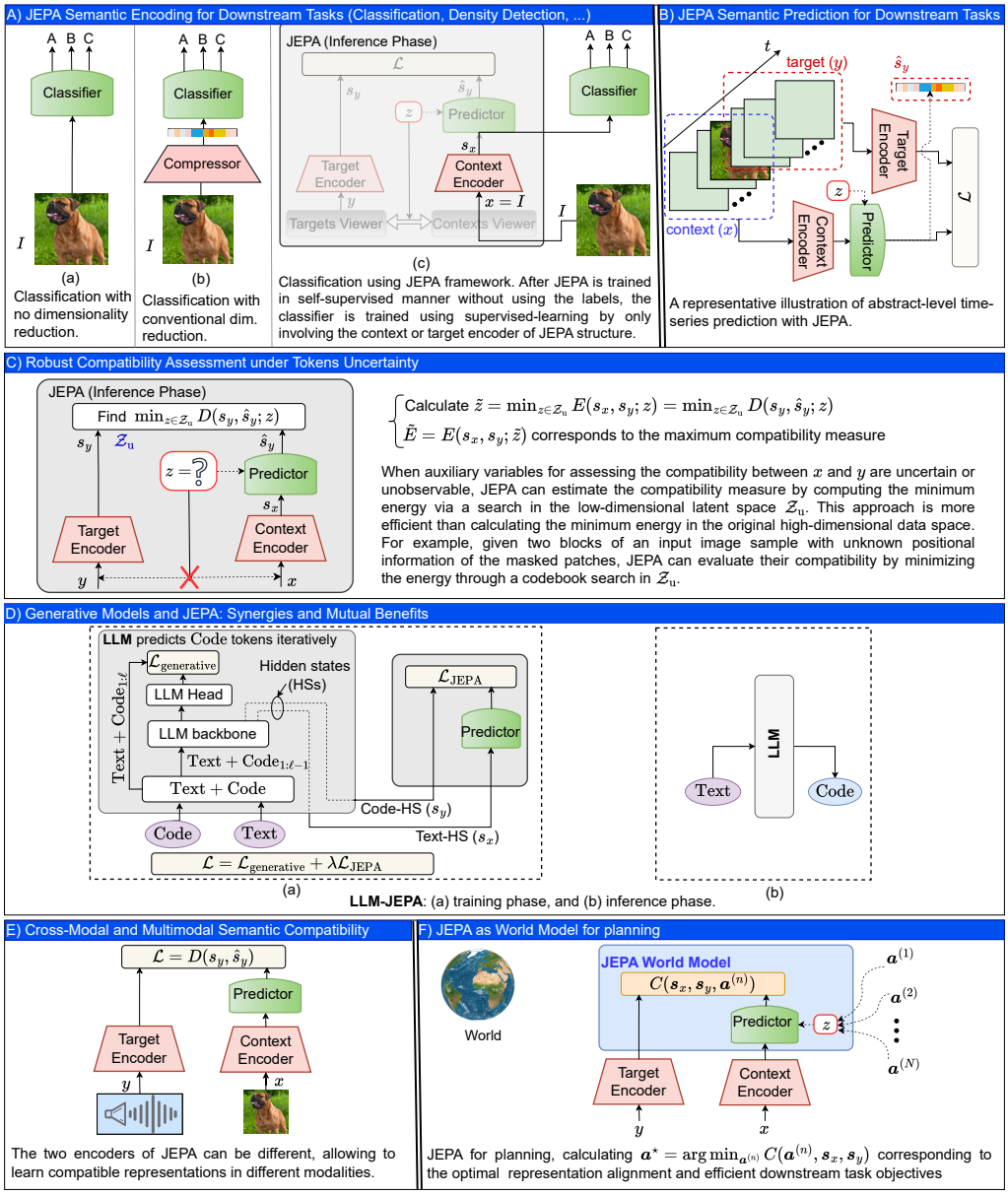

Fig. 9. Overview of representative JEPA applications across diverse tasks

(1) **Abstract-level pretraining:** Select a JEPA model and train or fine-tune it without using any of the available labels. If a pretrained backbone (such as I-JEPA, CNN-JEPA or SToP-JEPA for image datasets) is available, it can be adapted and finetuned to address the domain shift; otherwise, the model is trained from scratch on the dataset.

(2) **Feature extraction setup:** After training the JEPA model, all components except one of the context or target encoders are deactivated, and a shallow classifier (e.g., an MLP) is appended

to its output, as illustrated in Fig. 9A. The whole input is then fed directly to the context/target encoder, bypassing the context-target generation step used during pretraining.

(3) **Supervised classifier training:** Train the classifier using labeled data. Leveraging the context embeddings of JEPA, together with the corresponding available labels, a lightweight classifier (e.g., a simple two- or three-layer multilayer perceptron (MLP)) is trained in a supervised manner. During this stage, the context encoder parameters may be (i) frozen, (ii) partially fine-tuned by unfreezing upper layers, or (iii) fully fine-tuned jointly with the classifier, depending on the complexity-accuracy tradeoff considerations.

Extensive empirical studies have demonstrated the strong performance of this framework across a wide range of benchmarks [5, 53, 54, 64, 65]. In addition, the decoupling of representation learning from supervised classification provides further benefits in **label-limited regimes**: *available large unlabeled datasets can be exploited during JEPA training to capture semantic structure, while only the labeled subset is required for classifier training.*

3.1.2 JEPA for Density Detection. Although primarily designed for self-supervised predictive learning, JEPAs demonstrate remarkable "out-of-the-box" capabilities on downstream tasks beyond classification or detection. A key enabler of this lies in their *regularization and anti-collapse mechanisms*, originally introduced to preserve representation diversity and prevent mode collapse. Recent work by [7] reveals that leveraging JEPA cause the encoder $f_\theta(x)$ to *implicitly estimate the input data density* p_X, even though density modeling may not be an explicit goal. Note that the anti-collapse objective of JEPA encourages $f_\theta(x)$ to go toward producing Gaussian Embeddings (GEs). This property enables JEPA to provide the *approximation of data density without reconstructing* x *or requiring high-complexity generative training.*

To investigate how JEPA can be leveraged for density estimation, the authors of [7] focused on JEPA wherein the views are generated from stochastic transformations. The link between the density of (JEPA) encoded data $f_\theta(x)$ and the density of original data x is related to the Jacobian of the encoder $f_\theta(x)$, denoted by $Jf_\theta(x)$, and the spectrum of its singular values $\sigma_k(\cdot)$. When the JEPA objective reaches its optimum Gaussian distribution, regions of the input manifold that are compressed by the encoder (small singular values) correspond to areas of higher density, while expanded regions (large singular values) correspond to lower-probability configurations. Following mathematical derivations, the authors introduced the **JEPA-SCORE**, a log-scale Monte Carlo estimator of the data density, as:

$$\text{JEPA-SCORE}(x) = \sum_{k=1}^{\text{rank}(Jf_\theta(x))} \log \sigma_k\big(Jf_\theta(x)\big). \tag{9}$$

In practice, larger JEPA-SCORE values correspond to low-density samples, whereas smaller values identify samples occupying dense and well-represented regions of the training distribution. Fig. 10 illustrates this geometric interpretation: on the left, the input density p_X is mapped through the encoder f_θ to the latent density $p_{f_\theta(X)}$ on the right. Points with a small determinant of $(J_{f_\theta}(x_i)J_{f_\theta}(x_i)^\top)$ correspond to low JEPA-SCORE regions (blue, high-density), whereas points with a large determinant yield high JEPA-SCORE values (red, low-density). The diagram provides an intuitive visualization of how local volume changes encode variations in estimated data likelihood within the JEPA framework.

The implications of this finding are substantial. Since the JEPA-SCORE can be derived from any pretrained model without further fine-tuning, JEPAs naturally support data curation, outlier detection, and density estimation. In fact, regularization not only prevents collapse but also provides the model with a probabilistic awareness of its own training distribution. This property expands JEPA's role from a representation learner to a silent estimator of the data manifold itself. This

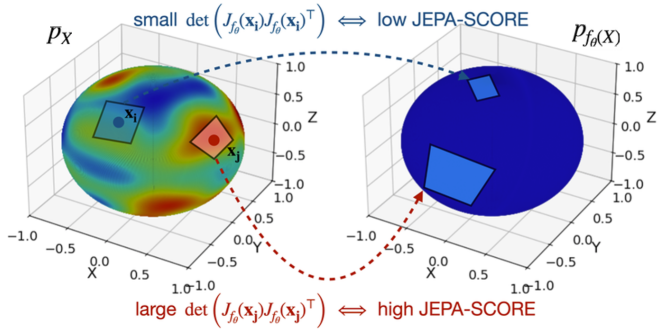

Fig. 10. Geometric intuition of the JEPA-SCORE [7].

provides insights for many applications. For instance, during inference, a very low JEPA-SCORE indicates that a sample lies far from the learned data manifold and can therefore be flagged as semantically anomalous.

3.2 JEPA Semantic Prediction for Downstream Tasks

A distinctive strength of JEPA lies in its ability to perform abstract-level prediction in various domains. For instance for *time-series* data, instead of forecasting pixels in video, waveform samples in audio, or raw sensor readings in control, JEPA predicts the *evolution* of latent embeddings that capture semantically meaningful structure. In the simplest way, for a time-series data $x[t]$, a history of length L (i.e., $x[t-L+1:t]$) can be defined as the *context view*, while the subsequent future samples with a horizon of length H (i.e., $x[t+1:t+H]$) constitutes the *target view*. The JEPA framework then learns to map the context view into predictions of the target embeddings $\hat{s}_y[t+1:t+H]$, in the latent space. This paradigm of forecasting time-series directly in the embedding space can enable applications across multiple domains. For instance, in audio modeling, A-JEPA has been proposed for predicting latent audio sequences [33]; in video and spatiotemporal data, V-JEPA variants have demonstrated strong performance in representation learning for computer vision and robotics [3, 11] enabling planning and reasoning; and in wireless communications, JEPA has been applied to predict temporal dynamics of channel state information (CSI) in the latent channel-chart domain [15, 17] to provide predictive pseudo-location representations of users.

In practice, JEPA can be leveraged for many representation-level prediction applications beyond the time-series domain. For instance, JEPA can be leveraged in semantic communication to only transmit the context tokens as well as auxiliary variables, letting the JEPA predictor positioned at the receiver to predict target tokens, providing the complete set of semantic representations. This idea will be further elaborated in Section 6.1.

3.3 Robust Compatibility Assessment under Uncertainties

Beyond observable conditioning variables such as positional indices of masked patches in I-JEPA, real-world systems often involve intrinsic stochastic factors associated with the dataset or model parameters that introduce uncertainty [56]. These *unobservable stochastic variables* cannot be directly exploited by the predictor as is the case with *observable conditioning variables*, but nonetheless influence the compatibility assessment between context and target embeddings. Instead of attempting to account for such uncertainty in the original high-dimensional input domain, JEPA offers a natural mechanism to incorporate it within the low-dimensional representation space

through its energy-based formulation [56, 57]. The uncertainties are introduced as latent variables $z \in \mathcal{Z}_{\mathrm{u}}$ that capture the effect of hidden factors originating from noise or stochasticity in the input domain. Compatibility between context and target embeddings is then assessed by incorporating the minimization of the energy function over $z \in \mathcal{Z}_{\mathrm{u}}$:

$$\tilde{z} = \arg\min_{z \in \mathcal{Z}_{\mathrm{u}}} E(s_x, s_y; z) = \arg\min_{z \in \mathcal{Z}_{\mathrm{u}}} D(\hat{s}_y, s_y; z), \tag{10}$$

resulting in the minimum energy

$$\tilde{E} = E(s_x, s_y; \tilde{z}). \tag{11}$$

Leveraging $\tilde{z}$ according to (10), the JEPA learnable parameters can then be trained to predict target representations from the context embeddings, as outlined in the previous section.

This strategy not only provides *robustness* by explicitly accounting for hidden stochastic perturbations in the context-target matching procedure, but also improves *efficiency* by resolving uncertainty in the compact latent domain $\mathcal{Z}$ rather than in the original input space. For example, JEPA can infer embeddings from noisy or occluded inputs by incorporating the most plausible latent configuration corresponding to the minimum energy during the inference phase. This can be performed through exploring a low-dimensional latent set $\mathcal{Z}$, making it far more efficient than resolving uncertainty directly in the high-dimensional input domain.

3.4 Generative Models and JEPA: Synergies and Mutual Benefits

Although JEPA is a non-generative framework, it can be contrasted and combined with generative paradigms that operate in the data or token domain. Here, the term *generative* broadly refers to observation-domain decoding, including both *reconstruction* of an input sample (e.g., autoencoding) and *new content generation* using models such as diffusion-based generative models, VAEs, or autoregressive generative models such as GPT-style LLMs. For instance, diffusion-based generative models synthesize new samples through iterative denoising in the observation domain, whereas autoregressive generative models generate outputs sequentially by factorizing the data or token distribution. Standard JEPA is different: it predicts target embeddings from context embeddings and minimizes a latent-space compatibility error, without requiring pixel-, waveform-, or token-level reconstruction. This distinction can improve efficiency and scalability for representation learning because JEPA avoids high-dimensional decoding and focuses on semantic predictability. However, this is also a limitation: standard JEPA alone is not designed for tasks requiring precise pixel-, waveform-, or token-level reconstruction. It should also be noted that although autoregression is widely used in LLM-based generative models, it is a general sequential prediction strategy and is not limited to token-domain generation. For example, autoregressive rollouts are also used in RL training and MPC-like planning with world models. Similar ideas can be used in non-generative sequential JEPA and JEPA-based world models, as discussed later in Section 4.2. Thus, in summary, JEPA can be complementary to generative models in complex tasks: JEPA is well suited for subtasks requiring compact predictive abstractions, while generative models are more suitable for subtasks requiring explicit high-fidelity synthesis or reconstruction in the original data domain. Two main paradigms can be considered for integrating JEPA and generative models.

3.4.1 Post-JEPA Generative Decoding. In this approach, a JEPA model is first trained using its standard predictive objective to learn context-aware latent embeddings. A generative *decoder* is then attached to the JEPA context/target encoder. During decoder training, the JEPA encoder can be kept frozen to preserve the learned invariant representation, or fine-tuned jointly to improve fidelity to the reconstruction task. This paradigm provides benefits such as *visualization insights*: decoding JEPA latents back to the data domain qualitatively reveals how semantics are preserved. For example, this has been utilized in several image-based related works such as I-JEPA [5] to

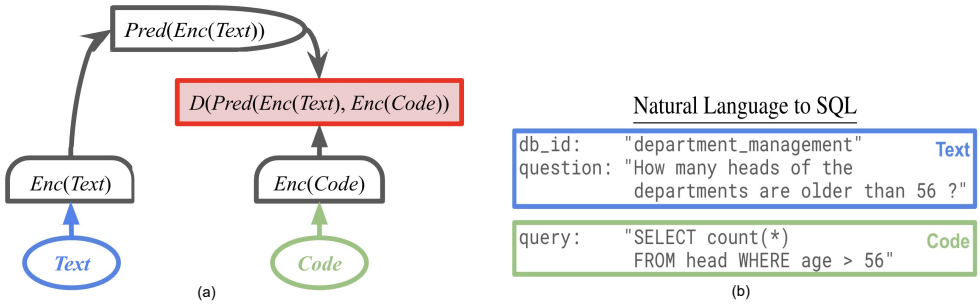

Fig. 11. (a) A representative Text-Code pair from Spider dataset where Text is the database ID and description of the SQL query, and Code is the SQL query itself. (b) Illustration of the JEPA block leveraged in the LLM-JEPA hybrid architecture proposed in [49].

show how reconstructions typically retain the important objects in a picture while suppressing non-semantic background/texture details, offering an intuitive diagnostic of JEPA's abstraction.

3.4.2 Joint Generative-Predictive Training Incorporating JEPA. A tighter integration jointly optimizes a generative reconstruction objective together with the JEPA predictive loss, leading to a composite formulation, typically represented as

$$\mathcal{L} = \mathcal{L}_{\text{generative}} + \lambda \mathcal{L}_{\text{JEPA}}, \tag{12}$$

where $\mathcal{L}_{\text{generative}}$ represents the desired data-domain loss (e.g., pixelwise mean-square reconstruction loss for images or next-token generation loss for language models), and $\mathcal{L}_{\text{JEPA}}$ is the latent-space predictive alignment loss. Optimizing (12) *balances* the generative capability and semantic alignment, with the trade-off controlled by λ. This architecture is particularly beneficial when two or more views of each data sample are either inherently available or can be readily generated to support the ultimate generative objective in the data domain. D-JEPA [18] provides a representative example of this joint generative-predictive paradigm. Its objective combines a diffusion/denoising loss $\mathcal{L}_d \equiv \mathcal{L}_{\text{generative}}$ with a JEPA-style prediction loss $\mathcal{L}_p \equiv \mathcal{L}_{\text{JEPA}}$. In this way, the diffusion component supports high-fidelity image generation/denoising, while the JEPA component preserves latent predictive structure. Leveraging the end-to-end loss $\mathcal{L} = \mathcal{L}_d + \mathcal{L}_p$, D-JEPA was reported to have superior scalability, faster convergence and better empirical performance than training with only the diffusion/denoising loss. A more detailed modality-specific discussion of D-JEPA is provided in Section 5.1.6.

Another representative instantiation is the **LLM-JEPA** architecture [49], which couples a JEPA encoder-predictor with a generative LLM. Given the tokens of a *Text*, as well as the first $\ell - 1$ tokens of a *Code* corresponding to the given Text (denoted by $\text{Code}_{1:\ell-1}$), the objective is to predict the next-token denoted by Code_ℓ in an autoregressive way, such that the overall Code corresponding to the given Text is finally generated. Here, the Text and Code are two views of the *same underlying knowledge*. For example, as shown in Fig. 11-a, the Text is an expression represented in natural language, while the Code is the standard representation of that expression in SQL query format. The main idea is that instead of accounting for a cross-modal attention of the LLM to couple Text and Code, delegate this task to JEPA to provide a separate, more efficient latent-alignment, as illustrated in Fig. 11-b; therefore the LLM simply targets the sequence modeling rather than accounting also

for a higher level of Text-Code coupling attention. The overall loss function is then

$$\mathcal{L}_{\text{LLM-JEPA}} = \sum_{\ell=1}^{L} \underbrace{\mathcal{L}_{\text{LLM}}\big([\text{Text} \,\|\, \text{Code}_{1:\ell-1}],\ \text{Code}_{\ell}\big)}_{\text{LLM: next-token on Code}} + \lambda \underbrace{D\big(\text{Pred}(\text{Enc}(\text{Text})),\ \text{Enc}(\text{Code})\big)}_{\text{JEPA: align Text and Code embeddings}}, \tag{13}$$

where $D(\cdot,\cdot)$ is a distance metric, and λ controls the trade-off. For the LLM, Text and Code tokens are concatenated into a single sequence, and an additive attention mask is applied to suppresses all cross-view (Text↔Code) attention between the tokens of Text and Code (offloading this task to JEPA), while preserving intra-view (Text↔Text, Code↔Code) tokens attention. The deactivation of attention between given tokens of a sequence is easily supported by most HuggingFace transformers. The proposed LLM-JEPA model has been trained over numerous datasets and various LLM models, where the results were shown to yield stronger cross-view alignment and improved performance over pure LLM baselines. The training and inference phase schematic architectures of LLM-JEPA are visualized in Fig. 9D.

A related JEPA-based integration is also explored in the JEP-KD framework [85], which extends the joint embedding predictive learning paradigm to visual-audio speech recognition. In this model, paired video sequences of lip movements and corresponding audio features serve as two complementary views of the same utterance, while the linguistic transcript is used as the target output for decoding. The architecture jointly trains a JEPA encoder-predictor module and a generative decoder within a three-stage procedure. The loss function is a composite objective that combines a reconstruction error term with a JEPA alignment term, along with other training objectives applied at different stages. Through this design, JEP-KD demonstrates how the JEPA paradigm can reinforce both representation learning and generative modeling in video-based speech understanding tasks.

3.5 Cross-Modal and Multimodal Semantic Compatibility Detection

Most existing JEPA applications employ similar structures for the context and target encoders, enabling compatibility detection and target prediction when both views originate from the same modality (e.g., images). More generally, however, the encoders can be designed to differ in the structure, allowing the context and target views to originate from distinct modalities [85, 92]. For example, in a video application involving paired image and audio sequences, one encoder may process visual frames while the other processes audio signals. Training JEPA on such data enables the model to learn abstract-level compatibility between the video and its associated audio track, making it possible to detect whether a given video segment is semantically aligned with the corresponding spoken content. Beyond *cross-modal* setups, JEPA can also involve *multimodal* applications, where both the context and target encoders receive inputs from multiple modalities simultaneously. This allows the model to learn joint representations that capture correlations across diverse input sources [58]. Such cross-modal and multimodal extensions open the door to applications such as video understanding, audio-visual correspondence, and multimodal anomaly detection. A review of existing works in this direction will be provided in Section 5.4.

3.6 JEPA World Models (WMs) for Planning

A powerful application of JEPA lies in its use as an internal or external world model for planning in agentic systems. Because JEPA learns a predictor that maps from a context representation s_x to future or masked target representations s_y entirely in latent space, a frozen JEPA can simulate the effect of hypothetical actions without reconstructing raw observations as visualized in Fig. 9F. An actor can therefore propose candidate actions and select the one that minimizes a JEPA-derived cost C, which can typically represent a combination of representation alignment error and/or

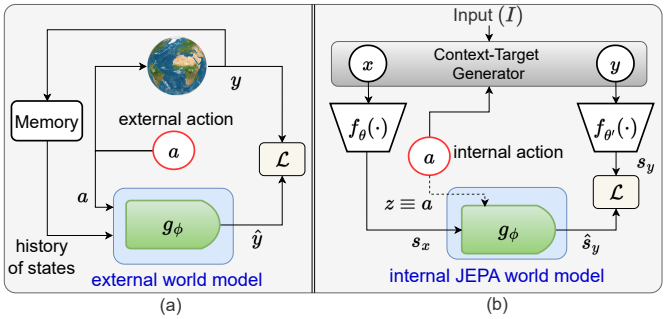

Fig. 12. External vs. JEPA-based internal world models. (a) A world model corresponding to a real external environment leveraged for agentic planning. (b) An abstract level JEPA internal world model.

downstream task objectives. This paradigm enables efficient action planning in a semantically rich, compact latent space [3, 83]. We explore the details in the following dedicated section.

4 Leveraging JEPA for Representing World Models

A world model is an internal mechanism that captures the system regularities to enable prediction, reasoning, and planning without real-world interaction. Building on fundamentals and applications of JEPA discussed in prior chapters, this section focuses on its role in world modeling.

Early world models predicted directly in the data domain (e.g., frame or pixel reconstruction) [14, 21, 23, 29, 39, 100], often within reinforcement learning or model predictive control frameworks. These approaches, reliant on raw sensory inputs, were computationally intensive and limited to short-term correlations with weak semantic encoding.

Subsequent *latent-dynamics world models* (e.g., [27, 39–42, 93, 96, 101]) transitioned to compact latent spaces, significantly improving computational efficiency and enabling long-horizon reasoning through simulated rollouts. These models, often integrated into reinforcement learning or model predictive control frameworks, relied on task-specific losses such as reconstruction errors or reward-based objectives to predict future states, which optimized their latent representations for control performance within particular environments (e.g., robotic navigation or video game dynamics). However, this focus tethered the latent variables to environment-specific dynamics, limiting their adaptability across diverse modalities (e.g., from visual context to audio targets) and their ability to capture intrinsic structural regularities independent of task-driven goals. This section describes how JEPA can be leveraged as world model enabling action planning.

4.1 Internal World Models Leveraging JEPA

Building upon the general JEPA formulation, we now examine how this architecture inherently functions as an *internal world model*, to capture predictive perceptual latent dynamics between context and target representations under internally defined actions or transformations. This perspective aligns with the representations of many JEPA variants reviewed in Section 5.

Fig. 12 contrasts the internal JEPA-based world model with its external, observation-domain counterpart. External world models, as illustrated in Fig. 12a, have been extensively studied as systems that learn the behavior of an *external* environment. In general, an external world model shown in Fig. 12a, is trained to predict next observations or states $\hat{y}(t+1)$ based on recent observations $y(t-L:t)$ and action $a(t)$ via a model g_ϕ, as

$$\hat{y}[t+1] = g_\phi(y[t-L:t], a[t]). \tag{14}$$

In contrast, Fig. 12b illustrates the paradigm in which the *JEPA predictor module* functions as an *internal world model*, simulating representation dynamics *within the agent* in the latent space, where actions here are not supplied by an external actor but are internal operations resulted from the view generation policy that modulate the relation between context and target representations. These *internal actions* are designed to expose the model to diverse and representative forms of variation, enabling it to learn latent context-target regularities. For instance, the Image World Model (IWM) [30] uses controllable transformations (rotation, brightness, color jitter) as internal actions, while recent extensions like V-JEPA 2 [3] and PLDM [83] support world models with multi-step latent rollouts for action planning.

It is seen from Figs. 12a and 12b that the internal JEPA world model and the external world model share a closely related predictive structure. In both cases, the evolution of states is governed by a model g_ϕ conditioned on an action variable a and a representation of the current state or recent states. For a generalized JEPA with temporal memory, the internal prediction can be expressed as

$$\hat{s}_y[t+1] = g_\phi(s_x[t-L:t]; a[t]), \tag{15}$$

where $a[t]$ denotes internal action parameters (e.g., target to context transformation type (e.g., rotation) and its corresponding parameters values (e.g. 30°) in image representation learning tasks such as IWM [30]) rather than external physical control signals. Comparing (15) with the external world model formulation in (14) shows that both embody same predictive principle: *predicting how state representations evolve under context/state profile and conditioning actions that drive change*. Finally, such internal latent predictive mechanisms can serve as a key component in foundational models for planning and control, as elaborated in the following.

4.2 JEPA World Models for Agentic Planning

In this section, we first outline LeCun's vision for an autonomous intelligence architecture centered on an external world model [56] and then show how JEPA enables an actor to perform planning using an *internal* world model. To elaborate further, let us initially consider a typical perception-action loop in LeCun's framework [56]. Two scenarios can be distinguished, denoted as *Mode-1* and *Mode-2* discussed below.

◦ **Mode-1**:*Reactive Perception-Action Loop Without World Model*: Fig. 13a depicts the standard reactive loop used in many end-to-end architectures. The perception module processes raw sensory input $x[t]$ into an abstract state representation $s[t] = f_\theta(x[t])$. This representation is directly fed to an actor policy π_ψ, which outputs action $a[t] = \pi_\psi(s[t])$ which in turn results in the next state observation $x[t+1]$.

◦ **Mode-2**: *Prediction and Planning using World Model*: Unlike Mode-1, where actions are selected reactively without any foresight, Mode-2 aims to achieve better performance by planning a sequence of upcoming actions based on predictions of future states of the world. This requires a world model capable of simulating the evolution of state representations in response to sequences of potential future actions. Inspired by Kahneman's *System 2* [52], LeCun's *Mode-2* view of autonomous intelligence leverages an operational pipeline where the actor, world model and cost evaluation module interact as illustrated in Fig. 13b. In this structure, given that a world model has been trained to simulate the environment outputs corresponding to the taken actions, the following steps are performed at each perception-action episode [56]:

Given a trained world model, in the most basic case, Mode-2 planning procedure for each time t can be summarized as the following procedure.

(1) *Perception encoding:* The current observation $x[t]$ is encoded into a latent state representation $s[t] = f_\theta(x[t])$.

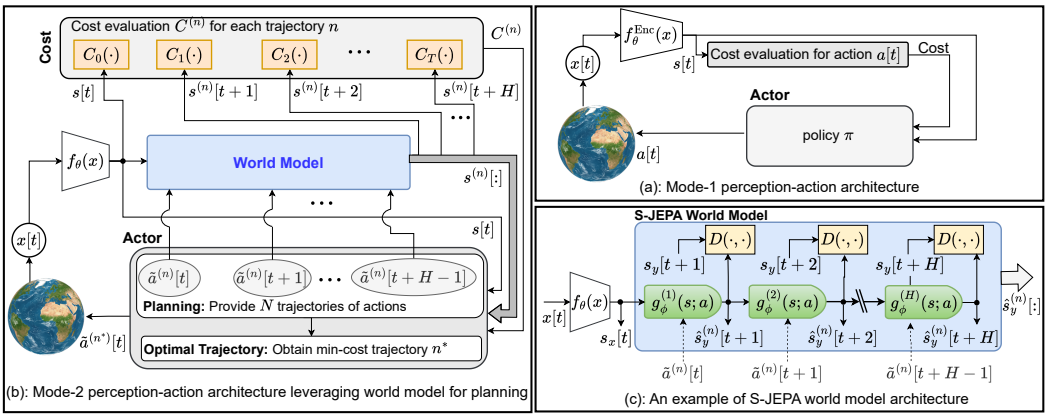

Fig. 13. Perception-action loop and incorporation of world model. (a) Mode-1, where decisions are made directly from observations with no action planning. (b) Mode-2, involving a world model and cost-based evaluation of simulated action trajectories. (c) An S-JEPA world model for planning with horizon of size H.

(2) *Recursive imagination of candidate futures:* For each candidate trajectory index n, the imagined rollout is initialized as $\hat{s}^{(n)}[t] = s[t]$. Two common rollout modes can then be distinguished. In *closed-loop* or *policy-guided* rollout, the action and latent state are generated in an interleaved autoregressive manner. At each imagined step $h = 0, \ldots, H-1$, an actor/policy π_ψ first proposes an action from the current imagined latent state,

$$a^{(n)}[t+h] = \pi_\psi\left(\hat{s}^{(n)}[t+h]\right), \tag{16}$$

and the world-model predictor then estimates the next latent state. For simplicity, we write the Markovian case , where the transition depends only on the current latent state and action:

$$\hat{s}^{(n)}[t+h+1] = g_\phi\left(\hat{s}^{(n)}[t+h]; a^{(n)}[t+h]\right). \tag{17}$$

Hence, the imagined trajectory evolves recursively as

$$\hat{s}^{(n)}[t] \to a^{(n)}[t] \to \hat{s}^{(n)}[t+1] \to a^{(n)}[t+1] \to \cdots \to \hat{s}^{(n)}[t+H]. \tag{18}$$

In *open-loop* planning, by contrast, a full candidate action sequence $\mathbf{a}^{(n)} = a^{(n)}[t : t+H-1]$ is sampled, initialized, or optimized before rollout, and the world-model recursion is then applied using these fixed candidate actions.

(3) *Cost evaluation:* The imagined latent trajectory $\hat{\mathbf{s}}^{(n)} = \hat{s}^{(n)}[t+1 : t+H]$ and its associated action sequence $\mathbf{a}^{(n)} = a^{(n)}[t : t+H-1]$ are evaluated by a planning cost

$$C^{(n)} = C\left(\hat{\mathbf{s}}^{(n)}, \mathbf{a}^{(n)}\right), \tag{19}$$

where $J(\cdot)$ may depend on the imagined states, the actions, or both, and may encode task reward, safety constraints, energy consumption, action smoothness, distance to a goal, or other task-specific criteria.

(4) *Trajectory refinement and selection:* The planner uses the evaluated costs to refine or select candidate futures. In sampling-based methods, a finite set of candidate trajectories is rolled out and evaluated where Steps 2–4 are repeated with new candidate trajectories. After the

final refinement round, the best trajectory can be selected as

$$\mathbf{a}^{\star} = \arg\min_{\mathbf{a}^{(n)}} C^{(n)}, \tag{20}$$

In gradient-based planning, one or more action trajectories are instead treated as optimization variables and updated by differentiating the cost through the rolled-out world model. In this case, Steps 2–4 are repeated until convergence or a fixed iteration budget is reached, and $\mathbf{a}^{\star}$ denotes the final optimized trajectory.

(5) *Acting:* The agent executes the first action, or the first few actions, of the selected trajectory $\mathbf{a}^{\star}$ in the real environment.
(6) *Buffering*: The resulting state transitions and associated costs are stored in short-term memory. These tuples may later be used for training purposes or adapting the critic.

Note that planning-based architectures using world models operate either in observation space [23, 29] (wherein these exists no observation encoding) or latent space [21, 41, 42, 76, 80].

The generic world-model planning scheme described above can be specialized to JEPA-based world models [3, 32, 69]. In this setting, the same agentic planning principle, i.e., optimizing action trajectories through imagined latent rollouts, is applied to action-conditioned JEPA predictors trained in representation space. A JEPA-based world model instantiates the recursive imagination procedure using the JEPA context–target prediction principle. In the temporal action-conditioned setting, the current observation $x[t]$ can be interpreted as the context view, while the next observation $x[t+1]$ acts as the target view. Their embeddings are

$$s_x \equiv s[t] = f_\theta(x[t]), \qquad s_y \equiv s[t+1] = f'_{\theta'}(x[t+1]), \tag{21}$$

and the action $a[t]$ plays the role of an auxiliary conditioning variable. The JEPA predictor is therefore trained as a latent transition model,

$$\hat{s}[t+1] = g_\phi(s[t]; a[t]), \tag{22}$$

by minimizing a latent prediction loss such as $D(\hat{s}[t+1], s[t+1])$, together with regularization terms that stabilize the representation space. As with other latent world models, JEPA world models can be trained from the past transition data tuples stored in an experience buffer, and may be combined with actor–critic or model-predictive control components depending on the planning architecture. Once trained, the predictor can be applied recursively in the imagination step above to generate latent rollouts without reconstructing future observations in the data domain.

This distinction is important: learning a JEPA world model means learning latent transition dynamics, whereas using it for planning additionally requires a criterion for selecting among imagined futures. Thus, JEPA world models can provide the predictive latent dynamics needed for planning, but the planning process still requires a cost, reward, value function, goal representation, or other criterion for selecting among predicted futures.

4.3 Recent JEPA-Based World Models

Before discussing planning-oriented JEPA world models, it is useful to note that some JEPA variants already instantiate a limited form of internal or perceptual world modeling. For example, the Image World Model (IWM) [30] extends image-based JEPA by conditioning the predictor on known transformation parameters, such as rotation, translation, or color changes. Unlike I-JEPA, which mainly predicts masked target embeddings from visible context embeddings, IWM learns how visual representations evolve under action-like transformations that builds target from the context, thereby capturing latent equivariant dynamics. However, these transformation variables are typically sampled or specified during view generation rather than optimized as control actions;

hence, IWM is better viewed as a transformation-conditioned internal JEPA world model, not as a full planning-oriented world model.

Recent works show a progression from JEPA as a static representation learner toward JEPA as a latent world model for prediction, reasoning, and planning. Seq-JEPA [32] extended JEPA from two-view prediction to sequential action- or transformation-conditioned latent prediction. Instead of predicting a single target representation from a context representation, seq-JEPA processes sequences of transformed observations and predicts the next latent state conditioned on the corresponding transformation. This makes seq-JEPA world-model-like, since it learns how representations evolve under known transformations or action-like variables. However, seq-JEPA is primarily proposed as a sequential representation-learning and latent-dynamics model, rather than as a complete goal-conditioned planner. Another relevant step toward planning-oriented JEPA world models is PLDM [83], an earlier JEPA-style latent dynamics model trained from reward-free offline trajectories. PLDM predicts future latent embeddings and uses VICReg-style regularization to prevent representational collapse. At test time, it performs goal-conditioned planning by optimizing actions so that the rolled-out latent trajectory approaches the encoded goal state. Thus, PLDM bridges sequential latent prediction and planning-oriented JEPA world models, although it relies on a more complex multi-term objective with several tunable loss weights.

For video and robotics, V-JEPA 2 and its action-conditioned variant V-JEPA 2-AC [3] extend the original V-JEPA framework toward physical-world prediction and planning. The base V-JEPA 2 model is pretrained in a self-supervised manner on large-scale video and image data to learn general spatiotemporal representations. To enable planning, V-JEPA 2-AC is then post-trained on robot interaction data by conditioning the latent predictor on robot actions. During inference, candidate action trajectories are rolled out in latent space and evaluated using goal-conditioned costs, such as distance to a visual goal representation, enabling model-predictive-control-style planning without pixel-level video generation or task-specific reward training. This demonstrates how JEPA-style latent prediction can support LeCun's Mode-2 planning paradigm through compact non-generative rollouts. V-JEPA 2.1 [69] improves this line mainly by strengthening the learned dense representations rather than by changing the basic goal-conditioned planning formulation. Compared with V-JEPA 2, it introduces a dense predictive loss where both visible and masked tokens contribute to training, applies deep self-supervision across intermediate encoder layers, and uses multimodal tokenizers for unified image/video training. These changes produce more spatially grounded and temporally consistent representations, improving dense vision, action anticipation, and planning-related performance.

Unlike V-JEPA 2 and V-JEPA 2.1, which primarily build on large-scale spatiotemporal video representation learning and then extend toward action-conditioned planning, the recent LeWorldModel (LeWM) [62] focuses on stable end-to-end training of a compact JEPA world model directly from raw pixels. LeWM is trained from pixel-based transition data, i.e., sequences of observations and actions $(x[t], a[t], x[t+1])$, by jointly learning an encoder and a latent predictor with only two loss terms: a next-embedding prediction loss and a Gaussian latent regularizer, which stabilizes the representation space without reconstruction. Compared with earlier end-to-end JEPA-style latent world models, LeWM reduces the number of tunable loss hyperparameters from six in PLDM [83] to one, making hyperparameter selection substantially simpler, e.g., through bisection search. With a compact model trained on a single GPU in a few hours, LeWM reports planning up to 48× faster than foundation-model-based world models while remaining competitive across diverse 2D and 3D control tasks. Once trained, LeWM predicts future latent embeddings and uses a lightweight Cross-Entropy Method (CEM)-style search over latent rollouts rather than training a separate heavy neural planner. In its goal-conditioned planning setting, candidate action sequences are selected by minimizing the latent distance between the predicted terminal representation and the desired

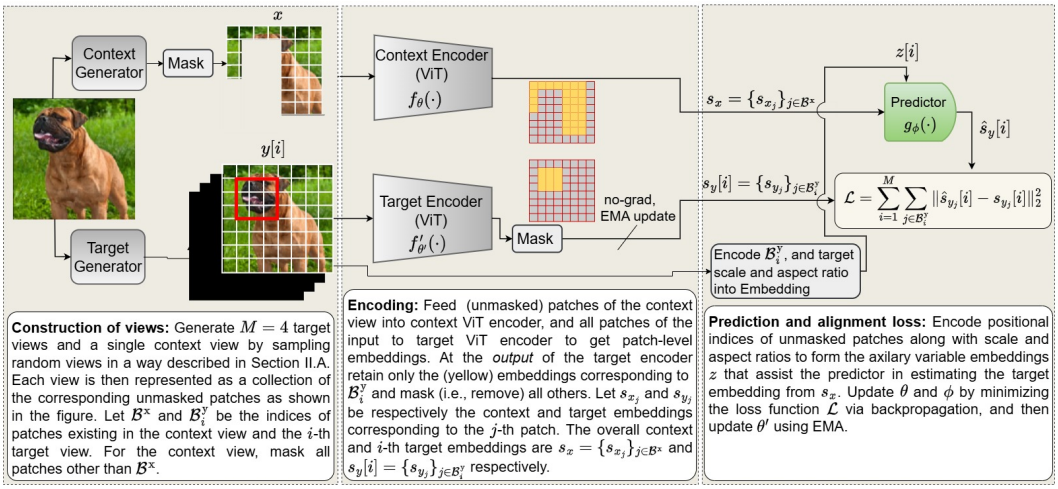

Fig. 14. The pipeline for training I-JEPA

goal representation s_g; equivalently, the rollout whose final predicted representation within the planning horizon is closest to s_g is selected as the optimal trajectory.

5 JEPA Across Modalities

In this section, we survey the literature on existing JEPA frameworks across data types including spatial, temporal, and spatiotemporal modalities. Here, *spatial* refers to 2D images and 3D point clouds, where pixels/points have explicit spatial coordinates; *temporal* denotes one-dimensional time series (e.g., sensor, audio or IoT streams); and *spatiotemporal* primarily covers video, in which spatial content evolves over time. We also review JEPA variants for other modalities (e.g., text) and cross-/multimodal settings, highlighting design choices and use cases.

5.1 Spatial JEPA for Images, Tabular Data, and 3D Point Clouds

5.1.1 I-JEPA. I-JEPA proposed and implemented by Meta AI [5], was the first adaptation of the JEPA framework to vision and established the template that many later variants were built upon. Fig. 14 summarizes its pipeline incorporating the three blocks stated in Section. 2.1.

Construction of views: Given an input image I, a single *context* view and M=4 *target* views are sampled. Concretely, I is partitioned into non-overlapping patches (16×16 patches on a 224×224 image, yielding $14 \times 14 = 196$ patch tokens). Blocks of patch indices are then sampled to define one masked context and M target blocks, following the mechanism explained in Section 2.1.1.

Encoding: The context and target views are encoded using the mechanism explained in Section 2.1.2, where both encoders are ViTs [22] composed of multi-head self-attention [90] followed by MLP blocks. The context encoder receives only the visible (unmasked) context patches, i.e., target masking is applied *before* encoding. The target encoder processes all patches (i.e., the *full* image); masking is applied *after* encoding so that only embeddings at target locations are retained.

Prediction and alignment loss: Context embeddings s_x are fed to a predictor as discussed in Section 2.1.3, where the predictor is narrow ViT. In addition to s_x, auxiliary metadata are also constructed by embedding target block's positional indices, as well as scale, and aspect ratio of each target to help the predictor more accurately predict target views. Conditioned on these, the predictor outputs the predicted embeddings for the target patches. The objective is the ℓ_2 distance between predicted and target embeddings, averaged over masked patches across all M target blocks,

as formulated in the right block of Fig. 14. The predictor and context encoder are trained jointly, while the target encoder is an EMA copy of the context encoder as discussed in Section 2.1.2.

Although I-JEPA has demonstrated strong capabilities in learning high-quality visual representations, several subsequent studies have proposed improvements and extensions to further enhance this architecture by mostly keeping its core structure, yet introducing new principles to further extend its capabilities, as presented in the following subsections.

5.1.2 MIM-JEPA. MIM-JEPA [91] exemplifies a hybrid CNN-ViT approach by introducing SCOTT (Sparse Convolutional Tokenizer for Transformers), a shallow convolutional stem that replaces the standard patch-and-embed tokenizer in a ViT. The goal is to inject local spatial biases early in the architecture to enhance data efficiency. In this model, both the context and target encoders are SCOTT-enabled ViTs, while the predictor is a shallow, standard Transformer. This design allows MIM-JEPA to significantly outperform the original I-JEPA on smaller datasets, providing a more computationally efficient and accessible implementation.

5.1.3 CNN-JEPA. Taking a different path, CNN-JEPA [53] fully embraces convolutional architectures by replacing the ViT backbone altogether. Here, the context and target encoders are standard CNNs, such as ResNet-50, and the predictor is also implemented as a lightweight, fully convolutional network. It similarly leverages the sparse convolution technique to effectively manage masked inputs and maintain the integrity of the predictive task. This pure CNN-based approach has also demonstrated superior performance and training efficiency compared to the ViT-based I-JEPA, particularly on small-scale benchmarks like ImageNet-100, further highlighting the advantages of CNN-native biases within the JEPA framework.

5.1.4 C-JEPA. The authors of [64] highlight two major shortcomings in I-JEPA: its EMA design fails to fully prevent collapse, and its predictor struggles to accurately model the mean patch representations. To address these issues, they present C-JEPA, which augments I-JEPA with VICReg regularization [12]. In this integration, the variance term keeps all embedding dimensions active with meaningful variation, preventing them from becoming useless; the covariance term reduces overlap by encouraging embedding dimensions to be independent; and the invariance term ensures that different augmented or masked views of the same image lead to similar representations.Notably, C-JEPA, just like I-JEPA, relies on the ViT architecture for both its encoder and predictor. This ensures that the observed performance gains come entirely from the proposed training and regularization strategies, rather than modifications to the underlying backbone. Experimental results on ImageNet-1K show that C-JEPA converges faster than I-JEPA and achieves higher accuracy.

5.1.5 SToP-JEPA. StoP-JEPA [10] tackles a key weakness in I-JEPA: its reliance on fixed positional embeddings makes the model overly dependent on knowing the exact location of image patches. This becomes a problem when spatial information is uncertain. Existing masked image modeling methods, such as MAE [44] and I-JEPA [5] predict masked tokens deterministically, not taking into account location uncertainty. For example, as shown in Fig. 15, given only part of a dog's image, locating its tail precisely is impossible. StoP-JEPA models each masked patch position as a Gaussian-distributed random variable whose mean is the original position and whose covariance is learned. By tying the noise projection matrix to the context projection matrix, the design prevents collapsing back to fixed positions and forces the predictor to learn representations that are less sensitive to exact location. This small change, requiring only a few extra lines of code, improves the learned representations from MIM, leading to better downstream performance on tasks such as ImageNet linear probing, as shown in Fig. 15, without adding computational cost.

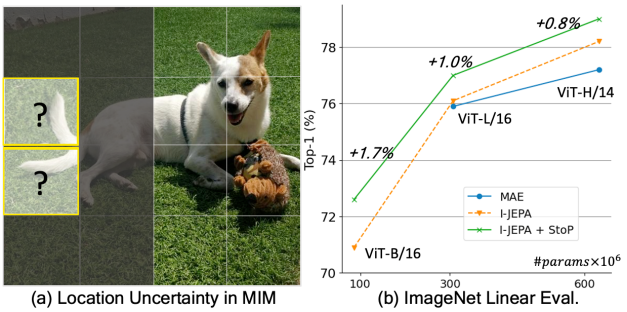

Fig. 15. SToP-JEPA [10]

5.1.6 D-JEPA. Denoising with JEPA (D-JEPA) [18] builds upon the foundations of I-JEPA. What sets D-JEPA apart is the addition of diffusion and flow matching losses along with an autoregressive generation process, turning JEPA from a representation learner into a true image generation model. The end-to-end loss function is formulated as the summation of a data-domain diffusion loss and a latent-domain JEPA loss. In practice, D-JEPA reaches state-of-the-art results on ImageNet conditional generation, producing high-quality samples with lower FID score and faster training. Beyond numbers, the model shows impressive flexibility, as it can naturally extend to image, video, and audio generation, providing a unified foundation for multimodal generative modeling.

5.1.7 Image World Model (IWM). IWM [30] extends image-based JEPA by replacing the standard split-view masking scheme with a transformation-conditioned view-generation process. Instead of sampling only disjoint context and target blocks as in I-JEPA, IWM generates paired views through known image transformations, such as rotation, translation, color changes, or masking. A source/context view and a target view are encoded by ViT-based encoders, and a transformer predictor, conditioned on the transformation parameters and mask tokens, predicts the target embedding from the source embedding. This enables the model to learn equivariant latent relationships, i.e., how image representations change under known transformations. From the modality perspective, IWM is hence an image-based JEPA that incorporates transformation-aware prediction, while its interpretation as transformation-conditioned internal JEPA world model is discussed in Section 4.3.

5.1.8 3D-JEPA. The authors of [48] introduced 3D-JEPA, the first extension of JEPA to 3D point-cloud data. Instead of forming masked patches as in image-based methods, 3D-JEPA follows the patch embedding strategy in [98], which builds point patches by first sampling M center points from the raw point cloud with farthest point sampling (FPS), then grouping each center's nearest neighbors k via KNN, and finally aggregating them with a lightweight PointNet [74]. Aiming to reduce target view overlap and enhance efficiency, 3D-JEPA proposes a masking approach in which target block centers are selected via FPS from the context set, followed by gathering nearby tokens within a fixed distance range. Standard transformers with self-attention layers are used for the encoders. The authors also introduce a context-aware predictor (called a decoder in the original paper) to enhance the encoder's ability to learn structural knowledge, continually feeding the encoded context into its layers while predicting target block representations. Compared to previous frameworks, 3D-JEPA produces semantically aligned and meaningful embeddings that better correspond to the objectives of the downstream task. It attains an average improvement of +31.43% accuracy across all three variants of the ScanObjectNN [97] benchmark. Furthermore, unlike previous SSRL methods [1], [72] that require 300 pretraining epochs, 3D-JEPA achieves superior performance using only 150 epochs, demonstrating both effectiveness and computational efficiency.

5.1.9 Point-JEPA. Point-JEPA [78] proposes a non-generative predictive architecture that learns high-level latent representations directly from spatially structured point tokens. A key innovation of this paper is the Greedy Sequencer, which arranges neighboring point patches into coherent spatial sequences, enabling the model to capture semantic continuity across local regions. Point cloud patches are generated using FPS and KNN grouping, where a mini-PointNet [74] encoder produces patch embeddings later processed by the JEPA framework. The architecture employs standard transformer-based encoders for both context and target streams, along with a predictor that aligns their feature spaces. Point-JEPA achieves state-of-the-art downstream performance with faster convergence, making it an efficient self-supervised solution for point cloud tasks with abundant unlabeled data and scarce labels.

5.1.10 T-JEPA. In [87] T-JEPA is proposed for structured tabular data, a domain where data augmentations for self-supervised learning are hard to design and often produce unrealistic samples. The authors adapt JEPA to tabular inputs to enable representation learning without any augmentation. Their method begins by tokenizing each feature column independently. Numerical features are normalized, and categorical features are embedded. Each column also receives index embeddings and type embeddings to capture its position and data type. The resulting tokens are processed by an FT Transformer [37] backbone, which is more suitable for modeling relationships between features than spatial grids. Training involves column-level masking, where one subset of features is used as context, while another non-overlapping subset is the target. In evaluations on several classification and regression benchmarks, T-JEPA produces richer and more task-relevant embeddings. As a result, deep tabular models can consistently match or outperform strong classical methods such as gradient boosted decision trees.

5.1.11 DSeq-JEPA. DSeq-JEPA [46] advances the I-JEPA framework by introducing a more human-like inductive bias through discriminative region selection and sequential latent prediction. Unlike I-JEPA, which predicts masked regions in a flat and parallel manner, DSeq-JEPA prioritizes regions based on their semantic importance. As shown in Fig. 16, the model uses an attention-driven saliency map to rank target regions instead of relying on random masking. This allows for a curriculum-like learning process. The model first learns to predict primary discriminative features, such as the face of an animal, and then moves to secondary details, such as background textures. This sequential progression from high-priority to low-priority cues results in richer latent embeddings. Consequently, DSeq-JEPA demonstrates superior efficiency and generalization across various downstream tasks, such as classification and object reasoning.

5.1.12 Other implementations of JEPA for spatial representation learning. Besides the aforementioned works, there exist several other studies applying the JEPA framework to images, focusing on improving its architecture and training process or extending it to more specialized tasks. DMT-JEPA [65] addresses a general weakness of I-JEPA that is its limited ability to capture fine grained local semantics. Instead of predicting masked patches independently, it constructs discriminative latent targets by aggregating features from semantically related neighboring patches through lightweight cross attention. This masked semantic neighboring strategy enriches local understanding. Mask-JEPA [54] integrates JEPA with mask classification architectures (MCA) to enable SSL directly on segmentation models. Its main idea is to use the transformer decoder of MCA as the predictor within the JEPA framework, allowing the model to jointly learn semantic representations and precise object boundaries, bridging pixel level decoding with latent space prediction. SparseJEPA [43] extends the JEPA framework by introducing sparse representation learning to make latent embeddings more interpretable and efficient. It adds a sparsity inducing penalty that groups semantically related latent variables in order to reduce redundancy in the embedding space. Table 1

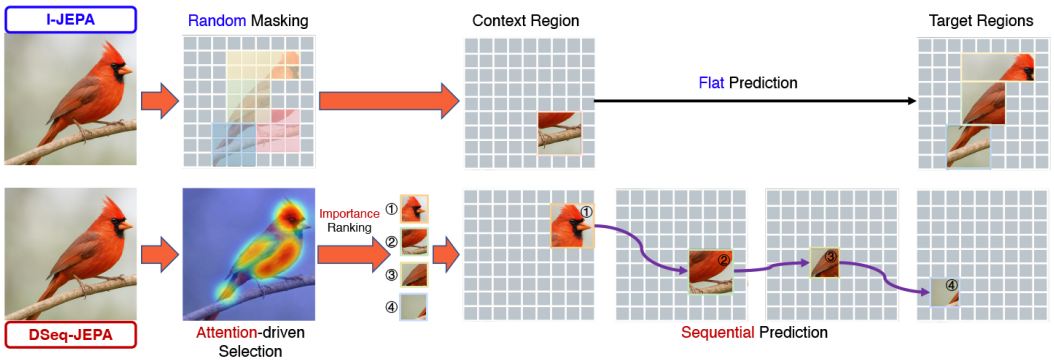

Fig. 16. DSeq-JEPA's sequential attention-guided prediction vs. I-JEPA's random parallel prediction [46].

summarizes representative JEPA-based frameworks across different modalities, including image, point-cloud, tabular, time-series, audio, video, multimodal, and world-model settings. Besides the image and point cloud data domains, JEPA can be leveraged to provide semantic representations for any other domain with spatially correlated data samples. For instance, the authors of [82] have utilized JEPA for graph representation learning, presenting the Graph-JEPA as a highly semantic self-supervised scheme to align context and target subgraphs in an embedding space.

5.2 Temporal JEPA for Time-Series and IoT Data

5.2.1 TS-JEPA for Time-Series Sensor Data. Girgis et al. [33] propose TS-JEPA, a JEPA variant tailored for high-dimensional time-series sensor data. The core idea is to predict temporal state transitions within a latent space rather than directly reconstructing raw sensor data. During the training phase, the devices transmit their high-dimensional states (originating from any type of time-series sensor data) alongside the control commands generated by the controller for those states to a central unit. The context encoder and predictor are jointly optimized to minimize the cosine distance between the predictor's forecasted embedding of a future state and the actual embedding of that state produced by the target encoder.

After training, each learned component is deployed where it best contributes to reducing network load: the context encoder runs on the devices to compress high-dimensional states into compact embeddings for transmission, while the predictor resides at the central controller to estimate future target embeddings based on the current context embeddings and the corresponding predicted control command. This design enables the controller to anticipate latent system dynamics without the need for continuous transmission of large raw data. In evaluation using time-series image data from an inverted cart-pole simulation, TS-JEPA achieved a substantial reduction in communication overhead while preserving high predictive accuracy and effective control performance in capacity-limited network environments.

5.2.2 Audio-JEPA. Audio-JEPA [88] extends I-JEPA to the audio domain by treating the spectrogram as a single-channel, potentially non-square image. An input waveform is first converted into a Mel-spectrogram [35] and partitioned into non-overlapping time-frequency patches, and then a fixed proportion of patches is randomly masked. The model employs a simple ViT backbone within its encoders: the context encoder processes visible patches, while a momentum updated target encoder processes the full spectrogram, including masked regions. A lightweight predictor aligns the context and target embeddings in latent space. Audio-JEPA demonstrates that this

straightforward mask-prediction objective can yield high-quality audio embeddings with far less pre-training data, making it an efficient and versatile approach for diverse audio tasks.

5.2.3 A-JEPA. Similar to Audio JEPA, the work [26] adapts the image-based I-JEPA framework to the audio domain by converting audio signals into Mel-spectrograms and treating them as two-dimensional representations with time and frequency axes. A key innovation lies in its masking strategy. Following [50], two masking patterns are used. The first is random block masking, which removes rectangular groups of patches that are relatively easy for the model to recover from. The second is time-frequency aware masking, which removes a portion of time and frequency, making the prediction task more difficult. Pre-training follows a curriculum masking schedule: the model starts with mostly random blocks and gradually shifts toward predominantly time-frequency aware masking. This progression enables the network to first learn from simpler contexts and then adapt to more complex patterns, fostering stronger semantic representations. In the fine-tuning stage, the context encoder is trained using Regularized Patch Masking. This means the model can't directly use the masked parts themselves, but must rely on information from the surrounding patches. This helps it stay robust and avoid overfitting. Thanks to this design, A-JEPA scales well and reaches top performance on large audio classification datasets like AudioSet-2M [31].

5.3 Spatiotemporal JEPA for Video

5.3.1 V-JEPA, V-JEPA 2 and V-JEPA 2.1. V-JEPA [11] extends JEPA to the video domain by predicting masked spatiotemporal representations in latent space rather than reconstructing pixels. The architecture is based on standard ViT backbones: the context and target encoders are full-sized ViTs, such as ViT-L or ViT-H, while the predictor is a narrower and lighter ViT. Each input video is first sampled as a 16-frame clip. A 3D convolutional stem converts the clip into spatiotemporal features, 3D positional embeddings are added, and the resulting tokens are flattened into a 1D sequence for ViT processing. A central design element of V-JEPA is 3D multi-block masking, which extends the block-wise masking strategy of I-JEPA to video. A 2D spatial mask is first formed from the union of several contiguous blocks and then applied consistently across all frames of the clip. This masks entire spatiotemporal regions and creates a challenging prediction task. With a masking ratio of about 90%, the context encoder observes only roughly 10% of the video tokens, while the predictor estimates the latent representations of the masked regions. The target features are produced by a momentum-updated target encoder with stop-gradient, which stabilizes training and helps prevent collapse. Empirically, V-JEPA learns transferable video representations, achieves strong performance on standard video representation benchmarks, and generalizes well to image classification despite being trained on video data.

Building on V-JEPA, V-JEPA 2 [3] and V-JEPA 2.1 [69] extend latent video prediction toward goal-conditioned decision-making, enabling their use as world models for MPC-like planning. A more detailed explanation of these two schemes provided in Section 4.3.

5.3.2 MC-JEPA. Bardes et al. propose MC-JEPA [13], a joint-embedding predictive architecture that integrates optical-flow estimation and content-based self-supervised learning within a shared encoder. Building upon the flow estimator architecture of PWC-Net [86] and the VICReg framework [12], the model simultaneously learns spatial correspondences between consecutive video frames and semantic representations of image content. This multi-task setup enables the flow estimation objective and the self-supervised learning objective to reinforce each other, resulting in visual features that encapsulate both appearance and motion cues. Trained on combined synthetic and real video datasets, the authors demonstrate that motion-aware self-supervision leads to richer and more transferable visual embeddings.

Table 1. Comparative summary of JEPA studies across various modalities

Modal	Model	Highlights (Innovation, Architecture, ...)	Context and Target Pair (x,y)	Year
Image	I-JEPA [5]	First adaptation of JEPA to image modality.	x: one random block of patches; y: multiple random blocks of patches.	2023
Image	MIM-JEPA [91]	Training ViT from scratch on small datasets using a sparse convolutional tokenizer (SCOTT).	x: patches remaining after masking; y: Random multi-block masking (masking ratio = 0.6)	2025
Image	CNN-JEPA [53]	Applying CNNs for JEPA rather than ViTs, enabling efficient pretraining with simpler architectures.	x: remaining patches after masking; y: union of multiple random blocks (as a single region).	2024
Image	C-JEPA [64]	Integrates JEPA with VICReg regularization to stabilize learning and prevent collapse.	x: a random block of patches; y: multiple random blocks of patches.	2024
Image	StoP-JEPA [10]	Introduces stochastic positional embeddings to address spatial uncertainty in masked image modeling.	x: a random block of patches; y: multiple random blocks of patches.	2023
Image	D-JEPA [18]	Integrates JEPA with diffusion and autoregressive generation for representation-generative modeling.	x: the remaining patches after masking; y: Random patch masking (variable masking ratio)	2024
Image	IWM [30]	Applies internal world-modeling using transformation-conditioned views for latent dynamics.	x: remaining patches after masking from the transformed source view; y: multiple random patch blocks from the transformed target view.	2024
Image	DSeq-JEPA [46]	Sequential latent prediction with human-like bias; uses attention-saliency for region ranking.	x and y selected from top-N saliency-ranked discriminative regions	2026
Image	DMT-JEPA [65]	enriches I-JEPA targets by aggregating semantically neighboring patch features via cross-attention, producing discriminative latent representations for masked prediction.	x: a random block of patches; y: multiple random blocks of patches.	2024
Image	Mask-JEPA [54]	use the transformer decoder of MCA as the predictor within the JEPA framework, allowing the model to jointly learn semantic representations and precise object boundaries.	x: patches remaining after masking; y: random masked patches based on preset masking ratio.	2024
Image	SparseJEPA [43]	JEPA augmented with sparse/grouped latent representations via a sparsity-inducing penalty that groups semantically related latent variables to reduce redundancy and improve interpretability.	x: a random block of patches; y: multiple random blocks of patches.	2025
Image (WM)	LeWM [62]	JEPA-based latent world model (WM) trained end-to-end from raw pixels with Gaussian latent regularization, used for planning.	x: current pixel state; y: next pixel state; predicts s_y from s_x, a for latent planning.	2026
3D	3D-JEPA [48]	Extends JEPA to 3D point clouds for the first time using PointNet-based patch embedding, FPS-based masking, and a context-aware predictor, improving ScanObjectNN accuracy with fewer epochs.	x: one random 3D block; y: a set of random 3D blocks, all extracted from the same point cloud via FPS + KNN.	2024
3D	Point-JEPA [78]	Uses a greedy sequencer that orders patch embeddings by proximity/index for efficient context-target selection, achieving classification accuracy of 93.7 ± 0.2% on ModelNet40.	x: one random 3D block; y: a set of random 3D blocks, all extracted from the same point cloud via FPS + KNN.	2025
Tabular	T-JEPA [87]	Adapts JEPA to structured tabular data; uses an FT-Transformer backbone and column-level masking to outperform classical GBDT methods in classification and regression.	x: a subset of unmasked feature columns; y: a non-overlapping subset of masked feature columns.	2024
Time-Series	TS-JEPA [33]	A JEPA-based framework for high-dimensional time-series sensor data that predicts future latent state transitions instead of reconstructing raw signals to reduce communication overhead	x and y are the current and the future system state.	2024
Audio	Audio-JEPA [88]	Adaptation of I-JEPA/JEPA to audio by treating Mel-spectrograms as image-like inputs. Comparable performance to wav2vec2.0 and data2vec with less than one-fifth of their pretraining data	x: patches remaining after masking; y: random patch masking (masking ratio = 0.4-0.6)	2025
Audio	A-JEPA [26]	Features Curriculum Masking (from Random Block to Time-Frequency Aware) for audio spectrograms	x: remaining patches after masking; y: spectrogram patches chosen by Curriculum Masking.	2023
Video	V-JEPA [11]	Adaptation of JEPA to video with a very high masking ratio (90%)	x: visible video patches; y: random spatial target blocks repeated across frames.	2024
Video	MC-JEPA [13]	Multi-task JEPA that jointly learns motion + content by coupling self-supervised optical flow (M-JEPA/PWC-Net style) with content SSL (VICReg) in a shared encoder.	x and y are obtained from paired inputs for flow and augmented image views for content.	2023
Video (WM)	V-JEPA 2 [3]	Extends V-JEPA with large-scale video/image pretraining and action-conditioned world-model (WM) (WM) post-training (V-JEPA 2-AC) for physical-world prediction and planning.	x: current video view; y: masked/future target view; V-JEPA 2-AC adds robot-action conditioning for latent rollouts.	2025
Video (WM)	V-JEPA 2.1 [69]	Enhances V-JEPA 2 with dense predictive loss and deep self-supervision for stronger video/image representation and planning.	x: visible/context image-video view; y: dense masked target view.	2026
Multi	TI-JEPA [92]	A text-image JEPA with that uses cross-attention to bridge the semantic gap between text and image features alignment.	x: combined masked image blocks and corresponding text; y: random multi-block masking.	2024
Multi	M3-JEPA [58]	An any-to-any multimodal JEPA using Multi-gate MoE to align diverse modalities in a shared latent space.	x: input modality features (e.g., Image); y: Output modality features (e.g., Text)	2025
Multi	JEP-KD [85]	Cross-modal distillation, using a noise-conditioned generative predictor to map lip-video features into audio-like embeddings.	x: video/lip sequence; y: audio/ASR encoder features.	2024

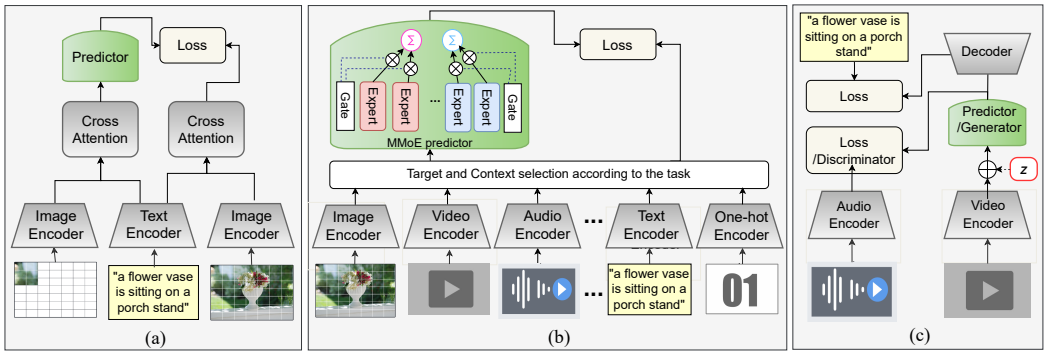

Fig. 17. Three multi-modal JEPA architectures: (a) TI-JEPA [92], (b) M3-JEPA [58], (c) JEP-KD [85]

5.4 Cross-Modal and Multimodal JEPA

5.4.1 TI-JEPA. TI-JEPA [92] offers a flexible joint embedding space for tasks such as multimodal sentiment analysis, with potential applicability to visual question answering and cross-modal retrieval. In the TI-JEPA pipeline, as shown in Fig. 17-a, both the target image and its associated context image are initially encoded by the image encoders. The corresponding text is processed separately by a text encoder. The resulting feature streams are then fed into cross-attention blocks to align textual and visual information. This interaction allows the model to project both text and image features into a joint embedding space, capturing fine-grained cross-modal relationships. After the cross-attention stage, the predictor module predicts the masked regions of the image, and then the reconstruction error is computed to update the energy function. TI-JEPA uses pretrained visual and textual encoder models that already contain rich, diverse representations. These pretrained components are frozen, and only the cross-attention modules and prediction layers are fine-tuned. This preserves the quality of existing features, prevents representational collapse, and focuses training on aligning the two modalities. Evaluation shows that TI-JEPA surpasses state-of-the-art relating works in terms of accuracy and F1-scores while retaining generalization across tasks.

5.4.2 M3-JEPA. M3-JEPA [58] (Fig. 17-b) is designed for true any-to-any multimodal learning. It is not locked into a single pairing and is capable of aligning almost any combination of modalities, such as text with image, image with audio, audio with text, and beyond. At its core, the model uses a sophisticated Multi-Gate Mixture of Experts (MMoE) [61] predictor. Each modality is first processed through a pretrained, largely frozen encoder, and the resulting embeddings are passed into this predictor. The predictor uses multiple gates to decide which expert should process the input, creating separate paths for information that is specific to each modality and for information that is shared between them. These paths are trained together with a combination of contrastive and regularization losses. This design allows the model to handle many different types of signals while still preserving what makes each modality unique. For training, M3-JEPA employs an Alternating Gradient Descent (AGD) method, as used in other multimodal studies [59], [2], which updates the predictor one task at a time, thereby avoiding the gradient conflicts that can slow or destabilize multitask learning. According to experimental results, M3-JEPA achieves competitive results across a range of multimodal benchmarks, shows robustness on unseen datasets, and remains computationally efficient, suggesting potential as a foundation for open-world self-supervised learning.

5.4.3 JEP-KD. JEP-KD [85] adapts the JEPA idea to visual speech recognition by reducing the semantic gap between visual lip movements and audio speech features. The model uses paired video and audio streams: a video encoder extracts visual features from lip movements, while a

pretrained audio speech-recognition encoder provides stronger audio representations as teacher features. Instead of directly forcing the video and audio embeddings to match with a simple distance loss, JEP-KD introduces a JEPA-inspired predictor/generator that maps video features combined with a noise vector z, toward an audio-like semantic representation. To make this cross-modal prediction more flexible, JEP-KD replaces the standard JEPA distance loss with an adversarial objective based on a discriminator that distinguishes real teacher audio features from generated audio-like features. A decoder (absent in standard JEPA) then maps the enriched representation to text for visual speech recognition. Training is organized in stages, including warm-up, adversarial prediction, and refinement, to stabilize the generator–discriminator interaction. This design shows how the JEPA predictor concept can be adapted beyond same-modality representation learning to cross-modal knowledge distillation.

6 Applications in Wireless Communications

Although JEPA has been extensively explored in various modalities, with notable progress particularly being achieved in computer vision, its application to wireless modalities and network-level tasks is relatively under-investigated. This section examines how JEPA has been adapted and employed in four representative domains of wireless communications systems.

6.1 JEPA for Semantic Communication in 6G Networks

As 6G moves toward autonomous systems and city-scale sensing, transmitting every bit of raw data is untenable. While 5G emphasizes ultra-reliability and massive connectivity, *semantic communication* reframes the objective: transmit *useful meaning* rather than pixel- or sample-accurate data [60, 81]. The core idea is to extract task-relevant, low-dimensional embeddings from high-dimensional signals and communicate only what matters for downstream utility, thereby reducing traffic and latency. In control-centric settings, this "prediction-anchored" view is exemplified by TS-JEPA [33], where device states are encoded into semantic embeddings, a predictor operates entirely in latent space to forecast future states, and a semantic actor maps embeddings to control commands; thus, communication focuses non-generative latents on *intent* and occasional *residuals* where prediction is insufficient, rather than streaming raw states.

In contrast, several proposed schemes such as TokCom [75] study *generative* semantic communication methods [75]. While JEPA can directly be leveraged to transmit the embeddings corresponding to the context and/or target blocks of the communication message in a non-generative scheme, in Section 7.2.4 we will also discuss how some existing token-based generative semantic communication can be extended JEPA-based semantic communication schemes.

6.2 Channel Charting

In wireless communications, Channel State Information (CSI) of antenna elements acts as a detailed fingerprint of the radio environment, reflecting fading, reflections, and user mobility [34]. Because channel characteristics evolve dynamically, learning and predicting their temporal behavior is essential for proactive resource allocation and mobility-aware control in 6G networks. Channel charting models aim to compress the high-dimensional CSI matrix into a compact 2D space representing the pseudo-location of the receiver at each time step. Traditional channel charting models rely on fixed statistical assumptions and fail to capture complex real-world dynamics, driving the need for data-driven approaches that learn latent channel evolution directly from observed CSI. Existing channel-charting techniques [28, 84] use self-supervision to construct static latent maps from raw CSI but lack predictive modeling of temporal evolution. To overcome this limitation, Bou Chaaya et al. [15] introduce W-JEPA, which extends channel charting with JEPA to enable forecasting of future channel representations in latent space. The encoder learns the charting

mapping, while the predictor captures the temporal dynamics of latent states driven by user motion. This approach predicts future embeddings without reconstructing raw CSI, substantially reducing computational cost. Experiments on real CSI datasets show that W-JEPA delivers up to twofold higher prediction accuracy than benchmarks for long-term channel dynamics.

6.3 Integrated Sensing and Communication

Integrated Sensing and Communication (ISAC) systems represent the next evolution of wireless networks, enabling simultaneous communication and environmental sensing through shared infrastructure [94]. In such networks, a base station, can *sea* the surrounding environment using the same radio signals used for data transmission, greatly improving environmental awareness while saving hardware, frequency, and energy resources. The ISAC model may include heterogeneous sensing modalities such as cameras, LiDAR, radar, infrared sensors, ultrasonic sensors, and RF signals from CSI. However, integrating these heterogeneous modalities remains difficult since each produces independent, high dimensional data, resulting in heavy communication load and reduced sensing efficiency. To overcome this limitation, the authors of [70] utilized JEPA to move multimodal integration and processing from the data domain into the semantic space. Each sensor, such as a camera or Wi-Fi device, encodes its observation into a compact latent embedding through a semantic encoder, which is transmitted to the base station. When one modality is missing or corrupted, the base station performs semantic inpainting, predicting the missing embeddings from the available ones. The completed embeddings are then forwarded to downstream applications. Experiments conducted on a multimodal dataset of camera images and Wi-Fi CSI confirmed the effectiveness of the proposed framework in terms of accuracy and lower transmission overhead.

6.4 Resource Allocation in Wireless Network

Besides the semantic alignment problem in ISAC recently investigated in [70] as described above, JEPA can benefit various wireless resource allocation problems. In this domain, only a few studies have been reported so far. The first example is the TS-JEPA framework discussed in Section 5.2.1, which utilized the JEPA to model the temporal evolution of networked control systems in latent space, considering a capacity limited wireless network. By predicting future device states and control commands semantically, TS-JEPA enabled proactive scheduling and adaptive bandwidth allocation, effectively reducing communication load while maintaining control stability. In another direction, the authors of [17] leveraged a novel architecture featuring two coupled JEPAs, namely *Control JEPA* and *Wireless JEPA*, to jointly model and predict the dynamics of both the control system (represented by pixels) and the wireless propagation environment (from CSI) in the latent space. This architecture then utilized a deep reinforcement learning to derive a control policy from latent control dynamics and a wireless transmission power predictor to estimate scheduling intervals based on latent CSI representation. The objective was then represented as minimizing the usage of radio resources by utilizing the coupled JEPA networks to imagine the device's trajectory in latent space. Simulation results on synthetic multi-modal data showed that the proposed scheme reduced the transmit power by 50% while maintaining the control performance.

7 Limitations, Challenges and Future Directions

This section highlights limitations of JEPA and outlines open research challenges and promising future directions across application domains.

7.1 Limitations and Failure Modes of JEPA

Despite its advantages for non-generative semantic representation learning, JEPA also faces several limitations, some of which have been noted throughout the preceding sections, where these limitations may be mitigated through careful considerations.

7.1.1 Auxiliary Variables and Complexity of Context-Target Generation. For many applications, JEPA performance is highly sensitive to context–target view construction scheme. If the target representation is too easy to infer, the model may learn shallow local correlations rather than semantic abstractions; if it is too weakly correlated with the context, the prediction task may become unstable or uninformative. Thus, optimizing the auxiliary variables such as masking ratio and masking positional information can be a complicated task especially when samples are high dimensional and complex.

7.1.2 Regularization Considerations. JEPA can suffer from representational collapse or low-diversity embeddings when the predictive loss is not paired with suitable stabilization mechanisms. Practical implementations therefore rely on stop-gradient operations, EMA target encoders, teacher–student asymmetry, variance/covariance regularization, or various statistical constraints. However, these mechanisms are architecture- and hyperparameter-dependent: insufficient regularization may fail to prevent collapse, while overly strong constraints may reduce transferability.

7.1.3 Training and Prediction Stability. Training stability becomes more challenging in large-scale, multimodal, and sequential/hierarchical JEPA variants. The encoders and predictor must evolve at compatible rates; otherwise, the predictor may chase rapidly drifting targets or learn from targets that are too static. In sequential JEPA and JEPA-based world models, small latent prediction errors may also accumulate over multi-step rollouts, degrading long-horizon forecasting.

7.1.4 JEPA-based World-Model Planning. Recent JEPA world models such as LeWM show that latent predictive architectures can support efficient, low-complexity planning compared with reconstruction-heavy pipelines. However, many JEPA planning formulations are naturally goal-conditioned: they assume a target representation and search for actions whose predicted latent rollout approaches it. This is effective when the goal is known but the path is unknown, but less direct for open-ended decision-making where the optimal target state is unspecified. Such cases may require additional reward/cost design, goal proposal, or intermediate subgoal generation.

7.1.5 Reconstruction Limitation. Standard JEPA is not designed for precise pixel-, waveform-, or token-level reconstruction. This is beneficial for compact semantic representation learning, but limits its direct use in high-fidelity synthesis or exact data-domain recovery, where a decoder, autoregressive generator, or generative components may also be required.

7.2 Open Challenges and Future Directions

7.2.1 Advanced JEPA World Model Architectures.

• *Distributed JEPA Architectures*: Current JEPA frameworks are typically centralized with co-located world-model components. However, many applications can benefit from latent predictive JEPA models in distributed architectures. For instance, in uplink wireless semantic communication networks, encoders can be deployed on user devices while predictors reside at edge/base stations, enabling coordinated inference and training under strict privacy and latency constraints. Distributed JEPA thus applies to both deployment and training: during deployment, encoder and predictor are split across network nodes; during training, split-learning JEPA or federated-learning JEPA schemes train

them collaboratively. Key challenges include designing scalable and informative distributed contexts/targets, synchronizing the world model across nodes, analyzing accuracy/complexity/latency trade-offs and ensuring robustness against node dropouts and adversarial/poisoned updates.

• *Combined S-JEPA/H-JEPA World Models*: To address long-horizon and complex planning tasks, future works can investigate a unified S-JEPA/H-JEPA architecture that combines multi-level abstraction: S-JEPA learns fine-grained, fast-changing representation dynamics, while H-JEPA captures slow-varying, high-level semantic dynamics critical for long-term planning and decision making. Moreover, since most existing systems pair world models with non-trainable actors, integrating trainable actors with such JEPA world models can amortize search, reduce latency, and improve planning and reasoning performance.

7.2.2 *Research Directions in Image/Video JEPA.*

• *Causality in V-JEPA*: Current video JEPAs capture correlations but lack explicit detailed cause-effect structure. In this regard the training can be augmented with *interventional pairs* and a sparse, time-lagged dependency map over tokens/slots. Given a context, we can form two targets: (i) the original one and (ii) an *interventional* target created by a localized manipulation (e.g., zeroing an object's velocity or altering friction inside a mask). The predictor can then be trained so that (a) latents for causal *descendants* of the intervention change consistently with the manipulated outcome (equivariance), while (b) non-descendant latents remain stable (invariance). In parallel, a lightweight lagged dependency matrix (sparse, with temporal lags) can be trained to capture which components influence others across frames. This yields a JEPA that supports counterfactual queries and is more robust under distribution shift.

• *Stochastic-Token V-JEPA with Spatiotemporal Uncertainty*: Inspired by StoP-JEPA for images, where randomized token positions improve robustness to spatial uncertainty, a video variant can be considered that models uncertainty in *both* space and time. Each tubelet can be represented by *random* centers/extents allowing temporal indices to *jitter or drop* (to reflect variable frame rate, motion blur, or desynchronization). Coupling spatial and explicit *temporal* uncertainty yields rollouts that are robust to occlusion and misalignment.

7.2.3 *JEPA for Digital Twin Architectures.*

Digital twins range from virtual replicas that mirror the current physical state ("what now") to predictive systems that answer counterfactual ("what if") and forecasting ("what next") queries. Since raw-sensory simulation is computationally costly, latent digital twins operating in compact representation spaces are closely aligned with latent world models for control and optimization. However, while world models mainly learn latent dynamics for decision-making, digital twins also require physical interpretability and partial reconstruction of observable variables. Thus, JEPA's non-generative predictive abstraction is efficient, but insufficient when physical-state estimation or visualization is required. Future research should explore hybrid JEPA–generative architectures that combine latent predictive alignment with decoding modules for estimating key physical variables while preserving JEPA's efficiency and semantic consistency.

7.2.4 *Research Directions in 6G Wireless Communications.*

• *Implementation of JEPA-Anchored Semantic Communication*: A rigorous end-to-end design and evaluation of JEPA-based semantic communication for 6G networks remains an open research direction. Existing token-based generative semantic communication frameworks, such as TokCom [75], employ tokenizers, codebooks, and generative foundation models or multimodal LLMs to select, transmit, and reconstruct semantic tokens. Although such schemes can exploit context reconstruction, they may incur high computation and latency for long token sequences, and require maintaining synchronized codebooks and generative priors across endpoints. A JEPA-anchored alternative would transmit only compact context embeddings/tokens, together with auxiliary

conditioning variables such as masking positions, and use a latent world model at the receiver to predict the missing target embeddings. This non-generative prediction in latent space could enable higher compression ratios and lighter receiver-side inference when exact pixel-level reconstruction is not required. Key open problems include designing such a JEPA-based framework for semantic communication optimized under various uncertainty sources such as non-ideal communication channel model, designing quantized latent representations and lightweight predictors, synchronizing the shared JEPA world model between transmitter and receiver, handling prediction errors, and developing hybrid schemes in which generative models such as TokCom reconstruct missing context tokens before a JEPA predictor infers the remaining target representations.

• *Semantic and Predictive CSI representation for Extremely Large-Scale Antenna Arrays (ELAAs)*: Recent works on leveraging JEPA for wireless network tasks [17, 33, 70] highlight its potential for estimation and resource management in communication systems. A promising application lies in channel estimation and prediction for ELAAs, where the CSI exhibits two-dimensional, image-like spatial structures [25, 68]. This structural analogy makes ELAAs a candidate for self-supervised, image-based representation learning frameworks such as I-JEPA. One such application is related to the CSI of ELAAs in the near-field propagation region [67]. A challenge for ELAAs in the near-field is the probability of *partial visibility* of the antenna array, where subsets of antenna elements become shadowed or blocked, resulting in missing useful CSI for those array regions. Addressing this challenge parallels the masked image modeling paradigm in computer vision, where portions of an image are intentionally removed and predicted from their surrounding context. In the ELAA case, the missing CSI segments can be modeled as masked targets, while the available elements serve as contextual input that can be leveraged for the prediction of the latent target masks.

8 Conclusions

This tutorial has provided a comprehensive overview of Joint Embedding Predictive Architectures (JEPA), covering their foundational principles, architectural variants, and wide-ranging applications. We studied JEPA's strengths across various downstream tasks, semantic predictive tasks, guidance of generative models via latent alignment, emerging applications in semantic communication and 6G networks, and its central role as a predictive world model enabling planning. An extensive survey of state-of-the-art JEPA implementations across spatial (images, point clouds), temporal (audio, time-series), spatiotemporal (video), and multimodal data further highlights the framework's remarkable versatility and performance.Despite the advances, important research directions remain: scaling JEPA to massive multimodal and decentralized foundation models, extending its capability to novel downstream tasks, incorporating explicit causal reasoning into the predictive objective, and building principled connections between non-generative JEPA and generative paradigms, among others. Successfully designing, validating, and deploying practical JEPA-based predictive world models in domains such as 6G semantic/goal-oriented communication, smart-grid control, autonomous robotics, and medical imaging will be decisive in transforming JEPA from a powerful representation-learning method into a core building block of future foundational AI models.

References

[1] Karim Abou Zeid, Jonas Schult, Alexander Hermans, and Bastian Leibe. 2023. Point2Vec for Self-Supervised Representation Learning on Point Clouds. In *German Conference on Pattern Recognition (GCPR)*.

[2] Hassan Akbari, Dan Kondratyuk, Yin Cui, Rachel Hornung, Huisheng Wang, and Hartwig Adam. 2023. Alternating gradient descent and mixture-of-experts for integrated multimodal perception. *Advances in Neural Information Processing Systems* 36 (2023).

[3] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. 2025. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985* (2025).

[4] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. 2022. Masked siamese networks for label-efficient learning. In *European conference on computer vision*. Springer.

[5] Mahmoud Assran, Ishan Misra, Julien Mairal, and Yann LeCun. 2023. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Gökhan BakIr. 2007. *Predicting structured data*. MIT press.

[7] Randall Balestriero, Nicolas Ballas, Mike Rabbat, and Yann LeCun. 2025. Gaussian Embeddings: How JEPAs Secretly Learn Your Data Density. *arXiv preprint arXiv:2510.05949* (2025).

[8] Randall Balestriero and Yann LeCun. [n. d.]. LeJEPA: Provable and scalable self-supervised learning without the heuristics, 2025. *URL https://arxiv. org/abs/2511.08544* 10 ([n. d.]).

[9] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2021. BEiT: BERT pre-training of image transformers. *arXiv preprint arXiv:2106.08254* (2021).

[10] Amir Bar, Florian Bordes, Assaf Shocher, Mahmoud Assran, Pascal Vincent, Nicolas Ballas, Trevor Darrell, Amir Globerson, and Yann LeCun. 2024. Stochastic positional embeddings improve masked image modeling. *arXiv preprint arXiv:2308.00566* (2024).

[11] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. 2024. V-JEPA: Latent video prediction for visual representation learning. (2024).

[12] Adrien Bardes, Jean Ponce, and Yann LeCun. 2022. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=xm6YD62D1Ub

[13] Adrien Bardes, Jean Ponce, and Yann LeCun. 2023. MC-JEPA: A joint-embedding predictive architecture for self-supervised learning of motion and content features. *arXiv preprint arXiv:2307.12698* (2023).

[14] Daniel Bogdoll, Yitian Yang, Tim Joseph, and J. Marius Zöllner. 2023. MUVO: A Multimodal World Model with Spatial Representations for Autonomous Driving. In *IEEE Intelligent Vehicles Symposium (IV)*. arXiv:2311.11762.

[15] Charbel Bou Chaaya, Abanoub M. Girgis, and Mehdi Bennis. 2025. Learning Latent Wireless Dynamics From Channel State Information. *IEEE Wireless Communications Letters* 14, 2 (2025). doi:10.1109/LWC.2024.3510943

[16] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*.

[17] Charbel Bou Chaaya, Abanoub M Girgis, and Mehdi Bennis. 2025. From Pixels to CSI: Distilling Latent Dynamics For Efficient Wireless Resource Management. *arXiv preprint arXiv:2506.16216* (2025).

[18] Dengsheng Chen, Jie Hu, Xiaoming Wei, and Enhua Wu. 2025. Denoising with a Joint-Embedding Predictive Architecture. *arXiv preprint arXiv:2410.03755* (2025).

[19] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*. PMLR.

[20] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[21] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. 2018. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems* 31 (2018).

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

[23] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. 2018. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568* (2018).

[24] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. 2021. Whitening for self-supervised representation learning. In *International conference on machine learning*. PMLR.

[25] Mohammad Amir Fallah, Mehdi Monemi, Mehdi Rasti, and Matti Latva-aho. 2024. Near-field spot beamfocusing: A correlation-aware transfer learning approach. *IEEE Transactions on Mobile Computing* (2024).

[26] Zhengcong Fei, Mingyuan Fan, and Junshi Huang. 2023. A-JEPA: Joint-embedding predictive architecture can listen. *arXiv preprint arXiv:2311.15830* (2023).

[27] Tuo Feng, Wenguan Wang, and Yi Yang. 2025. A survey of world models for autonomous driving. *arXiv preprint arXiv:2501.11260* (2025).

[28] Paul Ferrand, Maxime Guillaud, Christoph Studer, and Olav Tirkkonen. 2023. Wireless channel charting: Theory, practice, and applications. *IEEE Communications Magazine* 61, 6 (2023).

[29] Chelsea Finn and Sergey Levine. 2017. Deep visual foresight for planning robot motion. In *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE.

[30] Quentin Garrido, Mahmoud Assran, Nicolas Ballas, Adrien Bardes, Laurent Najman, and Yann LeCun. 2024. Learning and Leveraging World Models in Visual Representation Learning. *arXiv preprint arXiv:2403.00504* (2024). Image World Models (IWM).
[31] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio Set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE.
[32] Hafez Ghaemi, Eilif Muller, and Shahab Bakhtiari. 2025. seq-JEPA: Autoregressive Predictive Learning of Invariant-Equivariant World Models. *arXiv preprint arXiv:2505.03176* (2025).
[33] Abanoub M Girgis, Alvaro Valcarce, and Mehdi Bennis. 2025. Time-series JEPA for predictive remote control under capacity-limited networks. *arXiv preprint arXiv:2406.04853* (2025).
[34] Andrea Goldsmith. 2005. *Wireless communications*. Cambridge university press.
[35] Yuan Gong, Yu-An Chung, and James Glass. 2021. AST: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778* (2021).
[36] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020).
[37] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. *Advances in neural information processing systems* 34 (2021).
[38] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems* 33 (2020).
[39] Yanchen Guan, Haicheng Liao, Zhenning Li, Jia Hu, Runze Yuan, Guohui Zhang, and Chengzhong Xu. 2024. World models for autonomous driving: An initial survey. *IEEE Transactions on Intelligent Vehicles* (2024).
[40] David Ha and Jürgen Schmidhuber. 2018. World models. *arXiv preprint arXiv:1803.10122* 2, 3 (2018).
[41] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2020. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603* (2020).
[42] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruizhe Villegas, David Ha, Honglak Lee, and James Davidson. 2019. Learning Latent Dynamics for Planning from Pixels. In *International Conference on Machine Learning*. PMLR.
[43] Max Hartman and Lav Varshney. 2025. SparseJEPA: Sparse Representation Learning of Joint Embedding Predictive Architectures. *arXiv preprint arXiv:2504.16140* (2025).
[44] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
[45] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
[46] Xiangteng He, Shunsuke Sakai, Shivam Chandhok, Sara Beery, Kun Yuan, Nicolas Padoy, Tatsuhito Hasegawa, and Leonid Sigal. 2025. DSeq-JEPA: Discriminative Sequential Joint-Embedding Predictive Architecture. *arXiv preprint arXiv:2511.17354* (2025).
[47] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 5786 (2006).
[48] Naiwen Hu, Haozhe Cheng, Yifan Xie, Shiqi Li, and Jihua Zhu. 2024. 3D-JEPA: A Joint Embedding Predictive Architecture for 3D Self-Supervised Representation Learning. *arXiv preprint arXiv:2409.15803* (2024).
[49] Hai Huang, Yann LeCun, and Randall Balestriero. 2025. LLM-JEPA: Large Language Models Meet Joint Embedding Predictive Architectures. *arXiv preprint arXiv:2509.14252* (2025).
[50] Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. 2022. Masked autoencoders that listen. *Advances in Neural Information Processing Systems* 35 (2022).
[51] Ian Jolliffe. 2011. *Principal component analysis*. Springer.
[52] Daniel Kahneman. 2011. *Thinking, fast and slow*. macmillan.
[53] András Kalapos and Bálint Gyires-Tóth. 2024. CNN-JEPA: Self-supervised pretraining convolutional neural networks using joint embedding predictive architecture. In *2024 International Conference on Machine Learning and Applications (ICMLA)*. IEEE.
[54] Dong-Hee Kim, Sungduk Cho, Hyeonwoo Cho, Chanmin Park, Jinyoung Kim, and Won Hwa Kim. 2024. Joint-Embedding Predictive Architecture for Self-Supervised Learning of Mask Classification Architecture. *arXiv preprint arXiv:2407.10733* (2024).
[55] Diederik P Kingma, Max Welling, et al. 2014. Auto-encoding variational bayes.
[56] Yann LeCun. 2022. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review* 62, 1 (2022).

[57] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. 2006. A tutorial on energy-based learning. *Predicting structured data* 1, 0 (2006).

[58] Hongyang Lei, Xiaolong Cheng, Qi Qin, Dan Wang, Huazhen Huang, Qingqing Gu, Yetao Wu, and Luo Ji. 2025. M3-JEPA: Multimodal Alignment via Multi-gate MoE based on the Joint-Embedding Predictive Architecture. In *Forty-second International Conference on Machine Learning (ICML)*.

[59] Valerii Likhosherstov, Anurag Arnab, Krzysztof Choromanski, Mario Lucic, Yi Tay, Adrian Weller, and Mostafa Dehghani. 2021. PolyViT: Co-training vision transformers on images, videos and audio. *arXiv preprint arXiv:2111.12993* (2021).

[60] Kun Lu, Qingyang Zhou, Rongpeng Li, Zhifeng Zhao, Xianfu Chen, Jianjun Wu, and Honggang Zhang. 2022. Rethinking modern communication from semantic coding to semantic communication. *IEEE Wireless Communications* 30, 1 (2022).

[61] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*.

[62] Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. 2026. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312* (2026).

[63] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. 2000. Constrained model predictive control: Stability and optimality. *Automatica* 36, 6 (2000).

[64] Shentong Mo and Shengbang Tong. [n. d.]. Connecting joint-embedding predictive architecture with contrastive self-supervised learning. *Advances in neural information processing systems* 37 ([n. d.]).

[65] Shentong Mo and Sukmin Yun. 2024. DMT-JEPA: Discriminative masked targets for joint-embedding predictive architecture. *arXiv preprint arXiv:2405.17995* (2024).

[66] Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. 2023. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning* 16, 1 (2023).

[67] Mehdi Monemi, Sirous Bahrami, Mehdi Rasti, and Matti Latva-aho. 2025. A Study on Characterization of Near-Field Sub-Regions for Phased-Array Antennas. *IEEE Transactions on Communications* 73, 5 (2025), 2964–2979. doi:10.1109/TCOMM.2024.3483046

[68] Mehdi Monemi, Mohammad Amir Fallah, Mehdi Rasti, and Matti Latva-Aho. 2024. 6G Fresnel spot beamfocusing using large-scale metasurfaces: A distributed DRL-based approach. *IEEE Transactions on Mobile Computing* 23, 12 (2024).

[69] Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. 2026. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482* (2026).

[70] Takayuki Nishio, Cheng Chen, and Mehdi Bennis. 2025. A Semantic Inpainting Framework for Distributed Cross-Modal Integrated Sensing and Communication. In *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*. IEEE.

[71] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. DINOv2:: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023).

[72] Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. 2022. Masked Autoencoders for Point Cloud Self-Supervised Learning. In *Computer Vision – ECCV 2022, Part II*. Springer.

[73] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. 2016. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[74] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[75] Li Qiao, Mahdi Boloursaz Mashhadi, Zhen Gao, Rahim Tafazolli, Mehdi Bennis, and Dusit Niyato. 2025. Token communications: A large model-driven framework for cross-modal context-aware semantic communications. *IEEE Wireless Communications* 32, 5 (2025).

[76] Sébastien Racanière, Théophane Weber, David Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adrià Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. 2017. Imagination-augmented agents for deep reinforcement learning. *Advances in neural information processing systems* 30 (2017).

[77] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PmLR.

[78] Ayumu Saito, Prachi Kudeshia, and Jiju Poovvancheri. 2025. Point-JEPA: A joint embedding predictive architecture for self-supervised learning on point cloud. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE.

[79] Jürgen Schmidhuber and Daniel Prelinger. 1993. Discovering Predictable Classifications. *Neural Computation* 5, 4 (1993), 625–635. doi:10.1162/neco.1993.5.4.625

[80] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. 2020. Mastering Atari, go, chess and shogi by planning with a learned model. *Nature* 588, 7839 (2020).

[81] Hyowoon Seo, Jihong Park, Mehdi Bennis, and Mérouane Debbah. 2023. Semantics-native communication via contextual reasoning. *IEEE Transactions on Cognitive Communications and Networking* 9, 3 (2023).

[82] Geri Skenderi, Hang Li, Jiliang Tang, and Marco Cristani. 2025. Graph-level representation learning with joint-embedding predictive architectures. *arXiv preprint arXiv:2309.16014* (2025).

[83] Vlad Sobal, Wancong Zhang, Randall Balestriero, Tim G. J. Rudner, Kyunghyun Cho, and Yann LeCun. 2025. Learning from Reward-Free Offline Data: A Case for Planning with Latent Dynamics Models. *arXiv preprint arXiv:2502.14819* (2025).

[84] Christoph Studer, Saïd Medjkouh, Emre Gonultaş, Tom Goldstein, and Olav Tirkkonen. 2018. Channel charting: Locating users within the radio environment using channel state information. *IEEE Access* 6 (2018).

[85] Chang Sun, Bo Qin, and Hong Yang. 2024. JEP-KD: Joint-Embedding Predictive Architecture based knowledge distillation for visual speech recognition. *IEEE Open Journal of Signal Processing* (2024).

[86] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. 2018. PWC-NET: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[87] Hugo Thimonier, José Lucas De Melo Costa, Fabrice Popineau, Arpad Rimmel, and Bich-Liên Doan. 2024. T-JEPA: Augmentation-free self-supervised learning for tabular data. *arXiv preprint arXiv:2410.05016* (2024).

[88] Ludovic Tuncay, Etienne Labbé, Emmanouil Benetos, and Thomas Pellegrini. 2025. Audio-JEPA: Joint-Embedding Predictive Architecture for Audio Representation Learning. In *ICME 2025*.

[89] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008).

[90] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[91] Carlos Vélez-García, Miguel Cazorla, and Jorge Pomares. 2025. Escaping the Big Data Paradigm in Self-Supervised Representation Learning. (2025).

[92] Khang HN Vo, Duc PT Nguyen, Thong T Nguyen, and Tho T Quan. 2024. TI-JEPA: An Innovative Energy-based Joint Embedding Strategy for Text-Image Multimodal Systems. In *International Symposium on Information and Communication Technology*. Springer.

[93] Hang Wang, Xin Ye, Feng Tao, Chenbin Pan, Abhirup Mallik, Burhaneddin Yaman, Liu Ren, and Junshan Zhang. 2025. AdaWM: Adaptive World Model based Planning for Autonomous Driving. In *ICLR*. arXiv:2501.13072.

[94] Zhiqing Wei, Wangjun Jiang, Zhiyong Feng, Huici Wu, Ning Zhang, Kaifeng Han, Ruizhong Xu, and Ping Zhang. 2023. Integrated sensing and communication enabled multiple base stations cooperative sensing towards 6G. *IEEE Network* 38, 4 (2023).

[95] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. 2022. SimMIM: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

[96] Ziyang Yan, Wenzhen Dong, Yihua Shao, Yuhang Lu, Liu Haiyang, Jingwen Liu, Haozhe Wang, Zhe Wang, Yan Wang, Fabio Remondino, et al. 2025. RenderWorld: World Model with Self-Supervised 3D Label. In *IEEE International Conference on Robotics and Automation (ICRA)*. arXiv:2409.11356.

[97] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. 2016. A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics (ToG)* 35, 6 (2016).

[98] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. 2022. Point-BERT: Pre-training 3D point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

[99] Richard Zhang, Phillip Isola, and Alexei A. Efros. 2016. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*. Springer.

[100] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. 2024. OccWorld: Learning a 3D Occupancy World Model for Autonomous Driving. In *ECCV*. Springer. arXiv:2311.16038.

[101] Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, et al. 2025. World4Drive: End-to-End Autonomous Driving via Intention-aware Physical Latent World Model. In *ICCV*. arXiv:2507.00603.