High-Performance Self-Supervised Learning by Joint Training of Flow Matching

Kosuke Ukita
Kyushu Institute of Technology

Tsuyoshi Okita
Kyushu Institute of Technology

Abstract

Diffusion models can learn rich representations during data generation, showing potential for Self-Supervised Learning (SSL), but they face a trade-off between generative quality and discriminative performance. Their iterative sampling also incurs substantial computational and energy costs, hindering industrial and edge AI applications. To address these issues, we propose the Flow Matching-based Sensor Foundation Model (SenFlow), which jointly trains a representation encoder and a conditional flow matching generator. This decoupled design achieves both high-fidelity generation and effective recognition. By using flow matching to learn a simpler velocity field, SenFlow accelerates and stabilizes training, improving its efficiency for representation learning. Experiments on wearable sensor data show SenFlow reduces training time by 50.4% compared to a diffusion-based approach. On downstream tasks, SenFlow surpassed the state-of-the-art SSL method on all five datasets while achieving up to a 51.0x inference speedup and maintaining high generative quality. The implementation code is available at `https://github.com/Okita-Laboratory/SenFlow`.

1 INTRODUCTION

Diffusion models have recently emerged as a leading class of generative models, initially achieving remarkable quality in image generation and now driving breakthroughs across diverse applications like video generation (Zhang et al., 2024), time-series forecasting (Meijer and Chen, 2024), speech synthesis (Shen et al., 2024), and scientific domains such as drug discovery (Huang et al., 2024) and robotics (Shaoul et al., 2025).

A pivotal discovery fueling their success is that in generating high-quality data, these models incidentally learn powerful discriminative features, suggesting their potential as a next-generation framework for self-supervised learning (SSL) (Chen et al., 2025; Xiang et al., 2023).

However, applying diffusion models to representation learning presents a fundamental dilemma. Repurposing existing generative models (e.g., DDAE (Xiang et al., 2023)) yields representations not optimized for recognition tasks (Yang and Wang, 2023), while redesigning them for representation learning (e.g., l-DAE (Chen et al., 2025), SODA (Hudson et al., 2024)) often sacrifices their powerful generative capabilities. This stems from an inherent trade-off between generative and discriminative quality (Chen et al., 2025; Dombrowski et al., 2024), as the goal of faithfully reproducing fine details conflicts with learning representations that are invariant to task-irrelevant noise.

A promising solution is to decouple the architecture into a representation encoder and a generative network trained jointly. However, implementing this with diffusion models introduces another major challenge: the immense computational cost of their iterative training and inference processes. To overcome these dual challenges of the performance trade-off and high computational cost, we propose the Flow Matching-based Sensor Foundation Model (SenFlow). SenFlow is built on flow matching, a recent paradigm that achieves diffusion-like generative quality with a simpler and faster synthesis process. Our core idea is to integrate the decoupled recognition-generation architecture within this computationally efficient framework. By uniting architectural separation with computational efficiency, SenFlow achieves high generative and discriminative performance without compromise, creating a more versatile and powerful foundation model.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

The contributions of this paper are as follows:

- We propose SenFlow, a novel foundation model that directly utilizes the generative process of flow matching for representation learning.
- We introduce an efficient method for jointly training a representation encoder and a velocity field prediction network using the flow matching objective function.
- We demonstrate that our model significantly outperforms the state-of-the-art SSL baseline, SSL-Wearables, on human activity recognition tasks.
- Through a Text-to-Signal generation task, we show that the learned representations capture not only discriminative information but also generative structure.

2 RELATED WORK

2.1 Self-Supervised Representation Learning

SSL learns general-purpose feature representations by generating supervisory labels from the data itself, without the need for manual annotations. Modern SSL has been primarily driven by two major paradigms.

Masked Modeling Masked modeling involves training a model to predict a masked portion of an input sequence from its surrounding context. This approach revolutionized natural language processing with the introduction of BERT (Devlin et al., 2019) and has since achieved tremendous success in diverse modalities, such as with Masked Autoencoders (MAE) (He et al., 2022) in computer vision. The essence of this method is to compel the model to acquire an understanding of global context and semantic representations by solving the task of restoring local information.

Contrastive Learning Contrastive learning defines different augmentations of the same data as "positive pairs" and augmentations from different data as "negative pairs." The model is trained to maximize the similarity between positive pairs and minimize it between negative pairs in the representation space. Representative methods include MoCo (He et al., 2020), SimCLR (Chen et al., 2020), and DINO (Caron et al., 2021). This paradigm is particularly known for learning features that exhibit excellent linear separability in downstream tasks.

A detailed discussion on why masked reconstruction is suboptimal for 1D sensor data and how our flow matching objective addresses this limitation is provided in Appendix D.

2.2 Representation Learning Using Diffusion Models

The powerful data modeling capabilities of diffusion models have opened up a new research frontier in SSL. Approaches in this area can be broadly categorized as follows.

Reusing Pretrained Diffusion Models Pioneering work (Xiang et al., 2023) discovered that the intermediate representations of diffusion models pretrained for generative tasks unintentionally contain high-quality discriminative features. DDAE (Xiang et al., 2023) demonstrated a way to leverage existing models as zero-cost feature extractors. RepFusion (Yang and Wang, 2023) further proposed a sophisticated method that combines knowledge distillation and reinforcement learning to dynamically extract optimal features tailored to a specific task. Diffusion Classifier (Li et al., 2023) extended this to zero-shot classification tasks. However, since these approaches rely on existing models optimized for generation, their representations are not always optimal for downstream tasks.

Specialization and Simplification for Representation Learning An alternative approach is to rethink the model architecture itself for the purpose of representation learning. l-DAE (Chen et al., 2025) deconstructed and simplified diffusion models into their essential component, the Denoising Autoencoder, showing that the core of representation learning lies in denoising within the latent space. SODA (Hudson et al., 2024) aimed to engineer semantically disentangled representations by introducing an information bottleneck into the architecture and imposing specific self-supervised tasks. While these studies offer important insights, they tend to sacrifice the powerful generative capabilities of diffusion models in pursuit of discriminative performance. Ukita and Okita (2024) further investigated the use of diffusion model representations specifically for Human Activity Recognition on wearable sensor data, providing a direct precursor to the present work.

2.3 Flow Matching for Generative Modeling

To address the computational cost issues of diffusion models, particularly their slow sampling speed, flow matching (Lipman et al., 2023; Liu et al., 2023) has recently been proposed as a new paradigm for generative modeling. Flow matching directly learns the velocity field of an Ordinary Differential Equation (ODE) that defines a continuous flow from a simple probability distribution to the data distribution. This approach aims for efficient, high-quality data generation, poten-

tially in a single step, without the need for iterative sampling like in diffusion models. Conditional Flow Matching (Tong et al., 2024a,b) provided a computationally tractable objective function for training neural networks by targeting simple velocity fields conditioned on individual sample pairs, which simplifies the otherwise difficult problem of learning the average velocity field. However, research on flow matching to date has predominantly focused on high-quality data generation itself. The discriminative capabilities of the representations learned in the process have not been a central focus, and attempts to apply them to build foundation models remain, to the best of our knowledge, largely unexplored in the literature. This paper represents, to our knowledge, the first systematic investigation of this direction.

3 BACKGROUND

In Section 3, we provide an overview of two fundamental technologies that underpin our proposed method: Latent Diffusion Models and Flow Matching.

3.1 Latent Diffusion Models

In contrast to earlier diffusion models such as Denoising Diffusion Probabilistic Models (DDPM), score-based models, and Stochastic Differential Equation (SDE)-based models (Ho et al., 2020; Song et al., 2021), Latent Diffusion Models (LDMs) (Rombach et al., 2022) construct the diffusion process in a compressed latent space, using methods like Variational Autoencoder (VAE) to encode pixel-space data. This approach significantly improves both the quality and speed of generation. Furthermore, the Diffusion Transformer (DiT) (Peebles and Xie, 2023) replaced the commonly used U-Net backbone with a Transformer module, demonstrating enhancements in generation quality and scalability. In this work, we use DiT as our base architecture.

3.2 Flow Matching

Conditional Flow Matching (CFM) resolves the computationally challenging problem of learning a distribution's average velocity field by targeting simpler fields conditioned on individual sample pairs. This approach yields a computationally tractable objective function for training neural networks. In this framework, we first draw a source sample $x_0 \in \mathbb{R}^d$ from a prior distribution p_0 that is easy to sample from, and a target sample $x_1 \in \mathbb{R}^d$ from the training data distribution p_1. Next, using a time variable $t \in [0, 1]$, a point x_t on the probability path connecting these two points is constructed, most commonly via linear interpolation as shown in Equation (1).

$$x_t = (1-t)x_0 + tx_1 \tag{1}$$

The velocity field model v_θ is trained to predict the target velocity, denoted as $u_t(x_t|x_1)$, for any point x_t on this path. The CFM loss minimizes the squared error between the model's prediction and this target velocity, as expressed in Equation (2) (Lipman et al., 2024).

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t,x_0,x_1}\left[\| v_\theta(x_t, t) - u_t(x_t|x_1) \|^2\right] \tag{2}$$

4 METHODS

In Section 4, we detail SenFlow, a novel foundation model that utilizes flow matching. Our method follows the conventional two-stage process of self-supervised learning (Devlin et al., 2019; Caron et al., 2021), which involves pre-training on an unlabeled dataset and subsequent adaptation to downstream tasks. We build this framework using flow matching, but diverge from typical conditional generation approaches used in tasks like Text-to-Image, which often rely on large-scale annotated datasets. In contrast, to mitigate annotation costs, we pre-train our model by learning a representation-conditioned generation process that does not require any labels. The representations obtained through this process are then leveraged for downstream applications.

4.1 Pre-training

The training architecture of the proposed SenFlow during pre-training is shown in Figure 1. SenFlow is primarily composed of two components: a "Representation Encoder $f_\phi(\cdot)$" and a "Velocity Field Network $v_\theta(\cdot)$". In this paper, we specifically refer to the neural network that takes input data x_1 and outputs a representation vector $r \in \mathbb{R}^d$ as the Representation Encoder, as shown in Equation (3).

$$r = f_\phi(x_1) \tag{3}$$

The representation obtained from the Representation Encoder is fed as one of the conditions to the Velocity Field Network. Specifically, our work deals with conditional generation, producing target data x_1 from a condition c such as text. To this end, we first extend the CFM loss in Equation (2) to a framework for conditional generation. With the goal of learning the flow from a prior distribution p_0 to a target distribution $p_1(x_1|c)$ given a condition c, the CFM loss for conditional generation is given in Equation (4).

$$\mathcal{L}_{CFM,\ cond}(\theta) = \mathbb{E}_{t,x_0,(x_1,c)}\left[\| v_\theta(x_t, t, c) - u_t(x_t|x_1, c) \|^2\right] \tag{4}$$

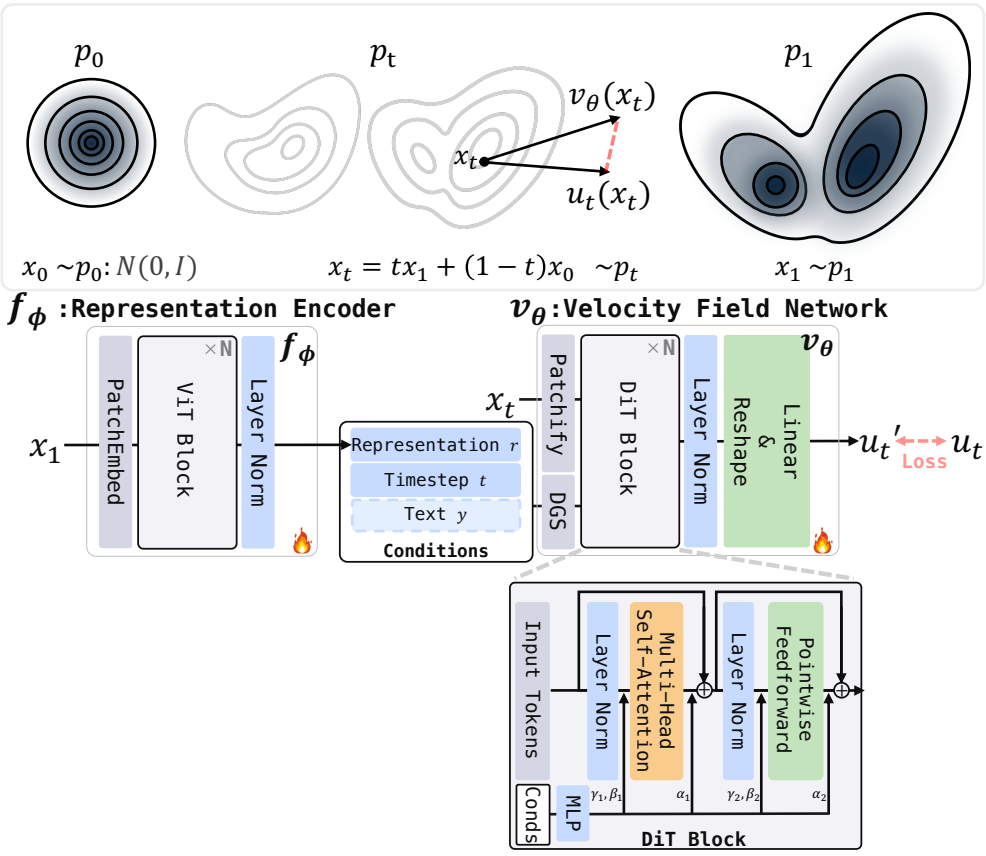

Figure 1: **Overall architecture of SenFlow during pre-training.** The model consists of (left) Representation Encoder f_ϕ and (right) Velocity Field Network v_θ. f_ϕ extracts representation r from input x_1 and supplies it as a condition to v_θ. v_θ takes point x_t on the probability path and conditions (r, time t, text y) and is jointly trained to predict the target velocity field u_t. The DGS module improves robustness by preventing over-reliance on the representation by randomly masking r during training.

The loss for SenFlow, which learns conditional generation conditioned on the active representation r obtained from the encoder, is shown in Equation (5).

$$\begin{aligned}&\mathcal{L}_{SenFlow}(\theta, \phi)\\ &= \mathbb{E}_{t,x_0,x_1}\left[\| v_\theta(x_t, t, r) - u_t(x_t|x_1, r) \|^2\right]\\ &= \mathbb{E}_{t,x_0,x_1}\left[\| v_\theta(x_t, t, f_\phi(x_1)) - u_t(x_t|x_1, f_\phi(x_1)) \|^2\right]\end{aligned} \tag{5}$$

The velocity field network v_θ is conditioned on the representation r, which depends on parameters ϕ, and is trained by optimizing with respect to parameters θ and ϕ. Although several methods have been proposed for training a representation-conditioned generation process in diffusion models (Rombach et al., 2022; Bordes et al., 2022), they all use representations from a pretrained, fixed encoder, making the conditioning information static and not a target of learning. In contrast, our method trains the Representation Encoder concurrently with the generation process. The representation is dynamically updated throughout training. By training the representation via gradients from the flow matching generation process, the encoder is expected to acquire not only generative capabilities but also advanced recognition abilities.

$$\begin{aligned}&\mathcal{L}_{SenFlow}(\theta, \phi)\\ &= \mathbb{E}_{t,x_0,x_1}\left[\| v_\theta^\phi(x_t, t) - u_t^\phi(x_t|x_1) \|^2\right]\end{aligned} \tag{6}$$

The SenFlow loss shown in Equation (5) can be rewritten as Equation (6), which offers the following interpretation: the parameters ϕ of the representation encoder f_ϕ do not merely provide a static condition, but dynamically parameterize both the model's predicted velocity field v_θ and the target velocity field u_t. This means that our framework does not solve a fixed, complex generation problem. Instead, ϕ actively designs a "path u_t^ϕ" that simplifies the problem itself, and the framework simultaneously performs this problem simplification and derives its solution. A detailed mathe-

matical analysis of the optimization dynamics of this joint training objective is provided in Appendix A.

Velocity Field Network The velocity field network in this study is based on the DiT (Peebles and Xie, 2023) architecture, which has demonstrated high performance in recent image generation tasks. However, since DiT is originally designed for 2D image data, we have modified the input part to process 1D sensor data. Specifically, we replaced the patch embedding layer at the input of DiT with a 1D convolutional layer. This converts the continuous time-series signal into a sequence of tokens that can be processed by the Transformer blocks. The configuration of the Transformer blocks follows the original DiT.

Representation Encoder Similarly, the representation encoder adopts a Vision Transformer (ViT) (Dosovitskiy et al., 2021) architecture. We replaced its patch embedding layer with a 1D convolutional layer to enable processing of 1D sensor data. It extracts a single feature vector from the input 1D sensor data, which serves as the representation vector.

4.1.1 Conditioning Mechanisms

In our method, we provide multiple pieces of information as conditions to the velocity field network to control and stabilize the generative process. During pre-training, the conditioning information consists of two types: (1) the representation vector r output from the representation encoder, which is the core of our proposed method, and (2) the timestep t. The timestep t is vectorized using a sinusoidal position encoding. These two condition vectors are combined element-wise and then integrated into each DiT block via *adaLN-Zero*[1]. Furthermore, considering various downstream tasks, this module is designed to allow for the addition of multiple arbitrary conditions, such as text prompts.

Dynamic Guidance Switching To enhance the model's versatility and improve performance on downstream tasks, we employ a strategy of intentionally masking the condition during training, which we name Dynamic Guidance Switching (DGS). Specifically, at each training step, the representation vector r is replaced with a zero vector with a certain probability (50% in this study). DGS, which is not used in the original DiT (Peebles and Xie, 2023) architecture and is our own design, enables a single model to learn both unconditional and conditional generation. It also acts as an implicit regularization on the representations acquired by the encoder. When the guiding information provided by the representation is masked, the generation process must maintain high generative capability autonomously without relying on that information. This prevents the generation process from becoming overly dependent on the representation and deteriorating in quality. Conversely, when the representation is provided as a condition, it must encode information that is genuinely essential for the generative process, thereby encouraging the encoder to capture compact and semantically meaningful features. Random masking thus acts as an implicit regularizer, preventing the encoder from producing degenerate representations that merely replicate low-level signal details. To verify this, we conducted a comparative experiment between training without DGS and with DGS applied, where the representation was randomly masked with a 50% probability. As shown in Table 1, the introduction of DGS consistently improved classification performance on the HAR task compared to not using it. This result demonstrates that masking the representation acts as a form of regularization, enabling the representation encoder to learn more robust and essential features. An extended sensitivity analysis across masking probabilities of 0%, 25%, 50%, and 75% is provided in Appendix C.

Table 1: **Effect of masking strategy**: Comparison of fine-tuning accuracy with 0% vs 50% mask probabilities. The 50% setting (DGS) consistently yields higher accuracy across all four datasets, demonstrating that random masking helps learn more robust representations.

Mask prob.	ADL	Opportunity	PAMAP2	REALWORLD
0%	.9370	.7775	.8868	.9004
50%	.9449	.7974	.8920	.9112

4.2 Downstream Tasks

4.2.1 Human Activity Recognition Task

To quantitatively evaluate the quality of the representations acquired through pre-training, we perform a Human Activity Recognition (HAR) task. This task utilizes the representation encoder trained during pre-training. We evaluate using two methods: transfer learning, where the pretrained encoder is frozen and only a linear classifier is trained, and fine-tuning, where all parameters are updated using the pretrained encoder as initialization.

4.2.2 Text-to-Signal Task

To verify that the representations acquired by SenFlow pre-training capture not just discriminative information but also richer semantic structures, we con-

[1]The conditioning mechanism employed in DiT.

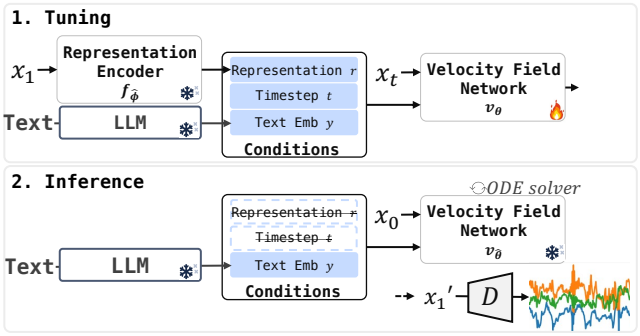

Figure 2: **Text-to-Signal workflow**: (1) In tuning, the network v_θ is fine-tuned conditioned on both representation r and text embedding y with frozen components. (2) In inference, the model generates signals x_1' from noise x_0 via an ODE solver, conditioned solely on text y.

duct a signal generation task conditioned on text descriptions. Signal generation in this task is based on the velocity field network trained during pre-training. While the generative model obtained from SenFlow pre-training is capable of both unconditional and representation-conditioned generation without additional fine-tuning, it does not inherently possess the ability to generate from text descriptions due to the self-supervised learning setup. Therefore, we tune the network to generate high-quality time-series signals conditioned on text descriptions, using the learned network weights as an initialization.

Figure 2 illustrates the workflow for tuning and signal generation during inference for the Text-to-Signal task. The tuning process is achieved by providing both the input text description and the representation as conditions. In diffusion models, it has been reported that incorporating representations into the conditional generation process improves generation quality (Rombach et al., 2022; Bordes et al., 2022). Therefore, we adopt an approach where we combine text embeddings with representations obtained from a fixed representation encoder and provide them as a condition. As shown in Table 2, this composite conditioning method consistently yields lower FID scores compared to using only text embeddings, demonstrating the generation of higher-fidelity signals. This result suggests that the representations acquired during pre-training richly capture the detailed statistical properties of the data and function as useful prior knowledge that further enhances the performance of the generative model when combined with high-level semantic information like text.

During inference, signals are generated solely from text descriptions. The text description input by the user is converted into a text embedding via an LLM, which serves as part of the condition for the velocity field network. Unlike the tuning phase, this is designed not to require a representation.

Table 2: **Impact of conditioning methods**: Comparison of generation quality (FID, Precision, Recall) on PAMAP2 and REALWORLD. Combining text with learned representations (Text + Rep) consistently outperforms Text-Only conditioning, demonstrating that incorporating representations significantly enhances signal fidelity.

Dataset	Conditioning	FID↓	Precision↑	Recall↑
PAMAP2	Text-Only	20.24	0.4843	0.5188
	Text + Rep	9.991	0.6300	0.6597
REALWORLD	Text-Only	12.33	0.2486	0.2480
	Text + Rep	8.261	0.6838	0.6925

5 EXPERIMENTS

In Section 5, we describe the datasets, experimental setup, and results. First, in Section 5.1, we explain the datasets used in our experiments. Next, in Section 5.2, we provide a quantitative evaluation of the quality of the learned representations through a human activity recognition task. In Section 5.3, we present a qualitative evaluation through a Text-to-Signal generation task conditioned on text descriptions. Finally, in Section 5.4, we report the results regarding the computational cost of SenFlow. To clearly demonstrate the effectiveness of our proposed SenFlow, we designed a diffusion model-based counterpart for direct comparison. We name this model the Diffusion-based Sensor Foundation Model (SenDiff). It employs the same joint training architecture of a representation encoder and a generative network as SenFlow, but with the generation mechanism replaced by a standard diffusion model. By directly comparing the performance and computational costs of SenFlow and SenDiff, we quantitatively evaluate the improvements in recognition performance and efficiency achieved by SenFlow.

5.1 Datasets

In this experiment, we use 3-axis accelerometer data from wrist-worn IMU[2] sensors. Following Yuan et al. (2024), we segment the signals into fixed-length sliding windows and treat them as independent inputs. All data used in the experiments are resampled to a frequency of 30 Hz using a 10-second sliding window. Details of the datasets are shown in Table 4. The pre-training data consists of the Capture-24

[2]Inertial Measurement Unit (IMU)

Table 3: **HAR classification performance**: Comparison of SenFlow against state-of-the-art baselines (e.g., SSL-Wearables) across five datasets. SenFlow consistently achieves the highest accuracy and F1-scores in both transfer learning and fine-tuning settings.

Method	ADL		Opportunity		PAMAP2		REALWORLD		WISDM	
	acc.	f1	acc.	f1	acc.	f1	acc.	f1	acc.	f1
Transfer Learning										
SSL-Wearables (Yuan et al., 2024)	-	.7540	-	.5470	-	.7250	-	.7710	-	.7230
1D-DINO (Caron et al., 2021)	.8835	.8691	.7557	.7682	.8111	.7981	.8124	.8243	.7747	.7716
SENvT-u4 (Okita et al., 2023)	.8394	.7936	.7103	.6826	.7157	.6912	.7470	.7512	.6667	.6618
SENvT-u7 (Okita et al., 2023)	.8472	.7849	.7150	.6909	.7098	.6869	.7628	.7692	.6819	.6766
SenDiff (ours)	.9055	.8813	.7787	.7760	.8171	.8090	.8614	.8741	.8261	.8224
SenFlow (ours)	.9370	.9168	.7881	.7792	.8467	.8373	.8683	.8832	.8337	.8322
Fine-tuning										
Random Forest[1]	.8346	.6830	.6814	.4260	.6602	.5467	.6618	.5712	.4630	.4224
Transformer scratch[2]	.8503	.8230	.6276	.6209	.7682	.7594	.8554	.8663	.8219	.8194
BERT (Devlin et al., 2019)	.8583	.8553	.7272	.7245	.7561	.7527	.7807	.7819	.7923	.7924
SSL-Wearables (Yuan et al., 2024)	-	.8290	-	.5950	-	.7890	-	.7920	-	.8100
1D-DINO (Caron et al., 2021)	.9024	.8780	.7656	.7741	.8341	.8223	.8878	.8978	.8693	.8684
SENvT-u4 (Okita et al., 2023)	.9087	.8780	.7489	.7411	.8488	.8454	.9015	.9124	.8693	.8679
SENvT-u7 (Okita et al., 2023)	.8992	.8684	.7429	.7448	.8477	.8445	.9058	.9163	.8875	.8867
1D-Diffusion Classifier (Li et al., 2023)	.8818	.8515	.7693	.7526	.8763	.8768	.8959	.8574	.8076	.8075
SenDiff (ours)	.9291	.9118	.7857	.7782	.8902	.8877	.9084	.9197	.8839	.8832
SenFlow (ours)	.9449	.9286	.7974	.7925	.8920	.8877	.9112	.9211	.8918	.8910

*1 Random Forest: 900-d input (concatenated x, y, z), depth 5, 10 trees.

*2 Transformer: depth 6, embedding dim 128, 4 heads, learning rate 0.001, 50 epochs.

dataset (Willetts et al., 2018), which we use as an unlabeled dataset. This dataset contains a total of 1,387,255 sliding windows. For the downstream tasks, we use five datasets: ADL (Bruno et al., 2013), Opportunity (Roggen et al., 2010), PAMAP2 (Reiss and Stricker, 2012), REALWORLD (Sztyler and Stuckenschmidt, 2016), and WISDM (Weiss et al., 2019).

Table 4: **Datasets**

Dataset	Subjects	Samples	Classes
Capture-24	152	1.3M	4
ADL	7	0.6K	5
Opportunity	4	3.9K	4
PAMAP2	8	2.9K	8
REALWORLD	14	12K	8
WISDM	46	28K	18

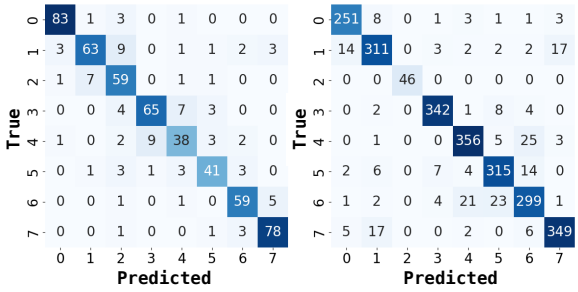

Figure 3: **Confusion matrices on HAR tasks**: The left panel displays transfer learning results on the PAMAP2 dataset, while the right panel shows fine-tuning results on the REALWORLD dataset. The strong diagonal patterns demonstrate SenFlow's high classification accuracy across diverse activity classes.

5.2 Human Activity Recognition Results

We conducted an HAR task to quantitatively evaluate the quality of the representations learned during pre-training. The Accuracy (acc.) and F1-score (f1) results are shown in Table 3. We compared our method against several baselines: SSL-Wearables (Yuan et al., 2024), a state-of-the-art multi-task SSL method for wearable data; 1D-DINO, a contrastive learning approach adapted from DINO (Caron et al., 2021); SENvT-u4 (Okita et al., 2023), a method employing multiple pretext tasks such as Transformer-based masking; its variant, SENvT-u7 (Okita et al., 2023), which extends this framework with additional tasks; 1D-Diffusion Classifier, an adaptation of Diffusion Classifier (Li et al., 2023); and our own SenDiff, a diffusion-based counterpart to SenFlow.

In both transfer learning and fine-tuning settings, SenFlow consistently outperformed all baselines across all five datasets. Under transfer learning, SenFlow improved upon the strong 1D-DINO baseline by up to +5.59% in accuracy and +6.06% in F1-score on REALWORLD and WISDM. The strong performance of

our diffusion-based SenDiff also validates the effectiveness of our decoupled architecture. In the fine-tuning setting, SenFlow demonstrated substantial gains over the SSL-Wearables baseline, with F1-score improvements ranging from +8.10% on WISDM to +19.75% on Opportunity. Figure 3 shows confusion matrices for selected results.

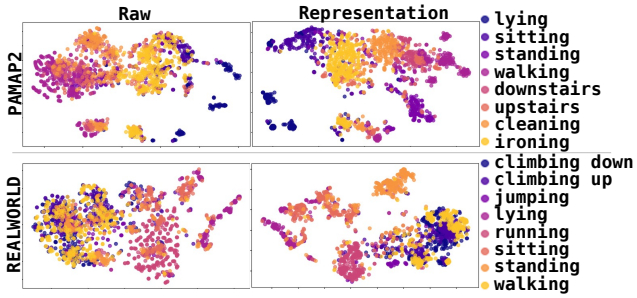

Figure 4: **t-SNE visualization of learned representations**: Comparison between raw data (left) and SenFlow representations (right) on PAMAP2 and REALWORLD datasets. The learned representations successfully disentangle activity clusters that overlap in the raw data space, demonstrating high-quality feature separation without supervision.

To visualize the quality and generalization ability of the learned representations, we used t-SNE (Van der Maaten and Hinton, 2008) to map signals from the out-of-domain PAMAP2 and REALWORLD datasets into a low-dimensional space (Figure 4). Despite being trained without labels, the representations form distinct clusters corresponding to activity classes. The effect is particularly clear on REALWORLD, where classes like 'running' and 'standing' form well-separated distributions, and on PAMAP2, where the representations disentangle clusters that overlap in the raw data space. To confirm the robustness of these results, we conducted evaluations across five independent random seeds; the mean F1-scores and standard deviations are reported in Appendix B.

5.3 Text-to-Signal Results

To verify that the learned representations capture rich semantic structures beyond discriminative features, we conducted a Text-to-Signal generation task. We constructed text prompts using metadata (e.g., age, sex) and activity labels from the PAMAP2 and REALWORLD datasets. These prompts were converted into condition vectors using the pretrained large language model Llama 3 (8B) (Grattafiori et al., 2024) in a modular text encoder designed for future extensibility. As shown in Figure 6, the generated signals clearly reproduce the characteristic patterns of specified actions (e.g., "walking," "lying"). Notably, the model captures subtle differences between similar activities, such as "walking" and "walking downstairs," and even appears to capture variations among individual subjects. This indicates an ability to translate semantic nuances from text into the statistical properties of time-series data. Furthermore, a t-SNE map of the PAMAP2 dataset (Figure 5) shows that the distribution of text-generated signals closely mirrors that of the original data, successfully distinguishing between both activities and subjects.

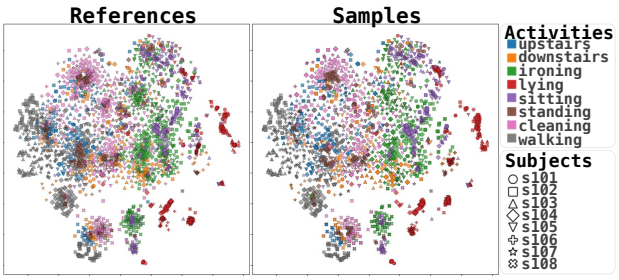

Figure 5: **t-SNE visualization of generated signals**: Comparison between real data (References) and signals generated from text conditions (Samples) on the PAMAP2 dataset. The generated distribution closely mirrors the real data, demonstrating that the model successfully captures distinct clusters for both activity types (colors) and individual subject characteristics (shapes).

These results qualitatively demonstrate our framework's ability to interpret complex semantic conditions in text to generate high-quality corresponding data. The success of this Text-to-Signal generation opens promising avenues for practical applications, such as synthetic data generation for simulations or conditional data augmentation.

5.4 Computational Cost Reduction

SenFlow led to a direct reduction in total energy consumption during the training phase. Through our experiments, we confirmed that flow matching training is more stable and converges faster than diffusion model training. This efficiency stems from its approach of directly learning a simpler velocity field via an ODE, bypassing the complex, iterative noising and denoising steps required by diffusion models. In a comparative experiment on training time with an equivalent setup (RTX 4090, batch size 1024), SenFlow (3.2h) achieved a training time reduction of 50.4% compared to SenDiff (6.5h). This indicates that the total energy consumed during the entire training process was halved, which is a significant contribution to the sustainability of large-scale foundation model development.

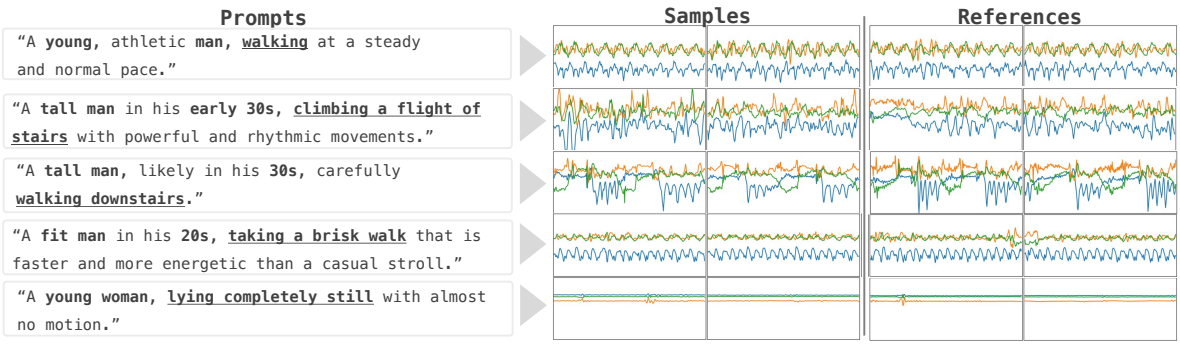

Figure 6: **Text-to-Signal generation examples**: Generated 3-axis accelerometer signals (Samples) conditioned on text prompts are shown alongside reference data (References). The model accurately translates semantic descriptions—such as activity type, intensity, and subject characteristics—into realistic time-series patterns, capturing distinct motion signatures like "walking" versus "climbing stairs."

Table 5: **Generation speed and quality during inference**: SenFlow achieves a 51.0x speedup compared to the diffusion-based SenDiff (1000 steps) while maintaining high generative quality (FID: 5.014), demonstrating significant efficiency gains.

Model	steps	time[s] / 1 sample↓	FID↓	IS↑
SenDiff	1000	4.62×10^{-2}	5.047	2.751
SenDiff	100	4.58×10^{-3}	8.837	2.800
SenDiff	50	2.29×10^{-3}	10.08	2.819
SenFlow	-	9.05×10^{-4}	5.014	2.626

In the inference phase, SenFlow showed a distinct advantage in generation speed. We evaluate generative quality using Fréchet Inception Distance (FID) (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016). As shown in Table 5, SenFlow achieved up to a 51.0x speedup over SenDiff while maintaining high generation quality (FID: 5.014). This speedup is due to the different generative processes: SenFlow solves an ODE, which a numerical solver can compute in far fewer steps than the numerous iterative denoising steps required by diffusion models. Furthermore, compared to generation with the faster DDIM[3] (50–100 steps), it achieved a speed improvement of 2.53–5.06x. This means that the energy consumption per generated sample was dramatically reduced without sacrificing generation quality. These efficiency gains highlight the potential of flow matching for real-time processing and edge AI applications, and can be further compounded by dedicated inference optimization techniques such as unified path classifier-free guidance (Ukita and Okita, 2026). From an energy perspective, substituting the Transformer-based encoder with a reservoir-based network (Kagiyama and Okita, 2025) represents a complementary direction for further reducing inference cost on resource-constrained devices.

[3]Denoising Diffusion Implicit Models (DDIM)

6 CONCLUSIONS

In this work, we proposed SenFlow, a novel foundation model that applies the generative capabilities of flow matching to representation learning. SenFlow aims to capture the essential structure of data while resolving the computational inefficiency of conventional diffusion models by jointly training a representation encoder, which extracts representations from input, and a flow matching model that generates data conditioned on those representations.

To validate the effectiveness of our proposed method, we conducted evaluations on multiple public datasets using wearable sensor data. The results show that in HAR tasks, SenFlow outperformed all existing SSL methods, including the state-of-the-art SSL-Wearables, across all five datasets. This result quantitatively demonstrates that SenFlow, based on its new principles, can acquire versatile representations applicable to a diverse range of downstream tasks. Furthermore, in terms of computational efficiency, SenFlow significantly reduced training time compared to the diffusion-based SenDiff, halving the total energy consumption. In the Text-to-Signal generation task, SenFlow achieved up to a 51.0x speedup over SenDiff while maintaining high generative quality, demonstrating that energy consumption can be dramatically reduced without sacrificing generative quality.

Based on these results, we conclude that SenFlow is a promising framework that not only overcomes the long-standing trade-off between generative quality and recognition performance but also enhances computational efficiency and generation quality. Future work includes applying this framework to diverse modalities such as images and audio and adapting it to more complex downstream tasks.

Acknowledgements

TO acknowledges the support from JSPS KAKENHI (grant numbers JP24K15107 and JP20K12065).

References

Bordes, F., Balestriero, R., and Vincent, P. (2022). High fidelity visualization of what your self-supervised representation knows about. *Transactions on Machine Learning Research*, pages 1–52.

Bruno, B., Mastrogiovanni, F., Sgorbissa, A., Vernazza, T., and Zaccaria, R. (2013). Analysis of human behavior recognition algorithms based on acceleration data. In *2013 IEEE International Conference on Robotics and Automation*, pages 1602–1607.

Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 9650–9660.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Chen, X., Liu, Z., Xie, S., and He, K. (2025). Deconstructing denoising diffusion models for self-supervised learning. In *The Thirteenth International Conference on Learning Representations*, pages 1–15.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Dombrowski, M., Reynaud, H., Müller, J. P., Baugh, M., and Kainz, B. (2024). Trade-offs in fine-tuned diffusion models between accuracy and interpretability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21037–21045.

Dong, J., Wu, H., Zhang, H., Zhang, L., Wang, J., and Long, M. (2023). Simmtm: A simple pre-training framework for masked time-series modeling. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 29996–30025.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, pages 1–21.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, pages 1–92.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 16000–16009.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 9729–9738.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, pages 1–12.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851.

Huang, L., Xu, T., Yu, Y., Zhao, P., Chen, X., Han, J., Xie, Z., Li, H., Zhong, W., Wong, K.-C., et al. (2024). A dual diffusion model enables 3d molecule generation and lead optimization based on target pockets. volume 15, page 2657. Nature Publishing Group UK London.

Hudson, D. A., Zoran, D., Malinowski, M., Lampinen, A. K., Jaegle, A., McClelland, J. L., Matthey, L., Hill, F., and Lerchner, A. (2024). Soda: Bottleneck diffusion models for representation learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23115–23127.

Kagiyama, M. and Okita, T. (2025). Knowledge distillation for reservoir-based classifier: Human activity recognition. *International Journal of Activity and Behavior Computing*, 2025(3):1–25.

Li, A. C., Prabhudesai, M., Duggal, S., Brown, E., and Pathak, D. (2023). Your diffusion model is secretly a zero-shot classifier. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2206–2217.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. (2023). Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, pages 1–28.

Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen, R. T., Lopez-Paz, D., Ben-Hamu, H., and Gat, I. (2024). Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, pages 1–83.

Liu, X., Gong, C., and liu, Q. (2023). Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, pages 1–33.

Meijer, C. and Chen, L. Y. (2024). The rise of diffusion models in time-series forecasting. *arXiv preprint arXiv:2401.03006*, pages 1–32.

Okita, T., Ukita, K., Matsuishi, K., Kagiyama, M., Hirata, K., and Miyazaki, A. (2023). Towards llms for sensor data: Multi-task self-supervised learning. In *Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing & the 2023 ACM International Symposium on Wearable Computing*, UbiComp/ISWC '23 Adjunct, pages 499–504.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. (2024). Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, pages 1–32.

Peebles, W. and Xie, S. (2023). Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 4195–4205.

Reiss, A. and Stricker, D. (2012). Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, pages 108–109.

Roggen, D., Calatroni, A., Rossi, M., Holleczek, T., Förster, K., Tröster, G., Lukowicz, P., Bannach, D., Pirkl, G., Ferscha, A., Doppler, J., Holzmann, C., Kurz, M., Holl, G., Chavarriaga, R., Sagha, H., Bayati, H., Creatura, M., and Millàn, J. d. R. (2010). Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, pages 233–240.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 10684–10695. IEEE Computer Society.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 2234–2242.

Shaoul, Y., Mishani, I., Vats, S., Li, J., and Likhachev, M. (2025). Multi-robot motion planning with diffusion models. In *The Thirteenth International Conference on Learning Representations*, pages 1–21.

Shen, K., Ju, Z., Tan, X., Liu, E., Leng, Y., He, L., Qin, T., Zhao, S., and Bian, J. (2024). Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. In *The Twelfth International Conference on Learning Representations*, pages 1–25.

Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al. (2025). Dinov3. *arXiv preprint arXiv:2508.10104*, pages 1–67.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, pages 1–36.

Sztyler, T. and Stuckenschmidt, H. (2016). On-body localization of wearable devices: An investigation of position-aware activity recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9.

Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. (2024a). Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, pages 1–34.

Tong, A. Y., Malkin, N., Fatras, K., Atanackovic, L., Zhang, Y., Huguet, G., Wolf, G., and Bengio, Y. (2024b). Simulation-free Schrödinger bridges via score and flow matching. In Dasgupta, S., Mandt, S., and Li, Y., editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 1279–1287. PMLR.

Ukita, K. and Okita, T. (2024). Analysis of human activity recognition by diffusion models. In *Companion of the 2024 on ACM International Joint Confer-*

ence on Pervasive and Ubiquitous Computing, UbiComp '24, page 458–463.

Ukita, K. and Okita, T. (2026). Efficient inference for flow matching via unified path cfg. In *Proceedings of the 30th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 1–12. Springer.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Weiss, G. M., Yoneda, K., and Hayajneh, T. (2019). Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access*, 7:133190–133202.

Willetts, M., Hollowell, S., Aslett, L., Holmes, C., and Doherty, A. (2018). Statistical machine learning of sleep and physical activity phenotypes from sensor data in 96,220 uk biobank participants. *Scientific Reports*, 8(1):1–10.

Xiang, W., Yang, H., Huang, D., and Wang, Y. (2023). Denoising diffusion autoencoders are unified self-supervised learners. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 15802–15812.

Yang, X. and Wang, X. (2023). Diffusion model as representation learner. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 18938–18949.

Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., and Xie, S. (2025). Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*, pages 1–43.

Yuan, H., Chan, S., Creagh, A. P., Tong, C., Acquah, A., Clifton, D. A., and Doherty, A. (2024). Self-supervised learning for human activity recognition using 700,000 person-days of wearable data. *npj Digital Medicine*, 7(1):1–18.

Zhang, H., Chen, X., Wang, Y., Liu, X., Wang, Y., and Qiao, Y. (2024). 4diffusion: Multi-view video diffusion model for 4d generation. In *Advances in Neural Information Processing Systems*, volume 37, pages 15272–15295.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. (2022). ibot: Image bert pre-training with online tokenizer. In *International Conference on Learning Representations*, pages 1–29.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   `The code to reproduce our main results in Table 3 is provided in the supplementary material. All datasets used are publicly available, as cited in Section 5.1.`
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   `We will provide a comprehensive list of all hyperparameters and their selection criteria in the supplementary material.`
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   `We report main results are single-run (Table 3); multi-seed evaluations are in Appendix B.`
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
   `The computing infrastructure (NVIDIA RTX 4090) is described in Section 5.4.`

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   `All public datasets and baseline models used in our experiments are cited throughout the paper, particularly in Section 5.`
   (b) The license information of the assets, if applicable. [Not Applicable]
   `The public datasets we used are standard academic benchmarks and do not have specific license restrictions that impact our work.`
   (c) New assets either in the supplemental material or as a URL, if applicable. [No]
   `We are not releasing any new datasets or models.`
   (d) Information about consent from data providers/curators. [Not Applicable]
   `We used publicly available datasets for which direct consent is not required.`
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [No]
   `The wearable sensor data used in this research is anonymized and does not contain personally identifiable information.`

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Appendix

A. Mathematical Details of Flow Matching and Joint Training

In the main text (Section 4.1), we introduced the joint training objective of SenFlow and presented the interpretation that the representation encoder f_ϕ dynamically parameterizes the target velocity field, effectively designing a "path" that simplifies the generation problem. A potential concern regarding this interpretation is as follows: because the probability path x_t is defined by standard linear interpolation $x_t = (1-t)x_0 + tx_1$, the target conditional velocity field $u_t(x_t \mid x_1) = x_1 - x_0$ is strictly determined by the boundary points and structurally independent of the encoder parameters ϕ.

In this appendix, we provide a rigorous theoretical reconciliation of this apparent contradiction. Drawing upon the theoretical foundations of Conditional Flow Matching (CFM) established by Lipman et al. (2024), particularly the *Marginalization Trick* and the standard properties of L^2 regression, we demonstrate that while the individual sample trajectories are indeed fixed, the *regression target manifold* optimized by the network is intrinsically parameterized by ϕ. This perspective mathematically validates our claim that the encoder actively modulates the tractability of the generative process, and provides a formal justification for why this joint training yields highly discriminative representations.

A.1. Review of General Conditioning in Flow Matching

To establish our analysis, we first review the generalized formulation of Conditional Flow Matching. According to the *Marginalization Trick* in Lipman et al. (2024), we can condition the probability paths and velocity fields on an arbitrary random variable $Z \in \mathbb{R}^m$ with probability density function p_Z. The marginal probability path $p_t(x)$ can be recovered by marginalizing over Z:

$$p_t(x) = \int p_{t|Z}(x \mid z)\, p_Z(z)\, dz. \tag{7}$$

Correspondingly, the marginal velocity field $u_t(x)$ that generates this probability path is given by the conditional expectation of the conditional velocity fields $u_t(X_t \mid Z)$:

$$u_t(x) = \int u_t(x \mid z)\, p_{Z|X_t}(z \mid x)\, dz = \mathbb{E}[u_t(X_t \mid Z) \mid X_t = x]\,, \tag{8}$$

where $p_{Z|X_t}(z \mid x)$ denotes the posterior density of Z given $X_t = x$ at time t. This formulation guarantees that if the neural network $v_\theta(x_t, t)$ is trained via the CFM objective

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,\, Z,\, X_t \sim p_{t|Z}(\cdot|Z)} \left[||v_\theta(X_t, t) - u_t(X_t \mid Z)||^2 \right] \tag{9}$$

then the optimal network $v^*_\theta(x_t, t)$ converges to the marginal velocity field $u_t(x) = \mathbb{E}[u_t(X_t \mid Z) \mid X_t = x]$. This equivalence—that the CFM objective and the FM objective share the same gradient with respect to θ—is established in Lipman et al. (2024) via the law of total expectation.

A.2. The SenFlow Joint Objective and the Conditional Target Field

In the SenFlow framework, the conditioning variable is not fixed to the target sample X_1, but rather to the representation vector extracted by the encoder, $Z = r = f_\phi(X_1)$. Examining the SenFlow objective under this lens gives

$$\mathcal{L}_{\text{SenFlow}}(\theta, \phi) = \mathbb{E}_{t,\, x_0,\, x_1} \left[||v_\theta(x_t, t, f_\phi(x_1)) - u_t(x_t \mid x_1)||^2 \right], \tag{10}$$

where $u_t(x_t \mid x_1) = x_1 - x_0$. One might correctly point out that $u_t(x_t \mid x_1)$ is constant with respect to ϕ, i.e., $\nabla_\phi\, u_t(x_t \mid x_1) = 0$.

However, analyzing the optimization dynamics from the perspective of the velocity network v_θ reveals a different picture. For any fixed encoder parameters ϕ, the network v_θ receives x_t, t, and $r = f_\phi(x_1)$ as inputs. By the standard property of the L^2 regression problem, the global minimum with respect to θ, in the limit of infinite network capacity, is the conditional expectation of the target given the inputs:

$$v_{\theta^*}(x, t, r) = \mathbb{E}[u_t(X_t \mid X_1) \mid X_t = x,\ f_\phi(X_1) = r] \ \triangleq\ u_t^\phi(x, r), \tag{11}$$

where X_t and X_1 denote the corresponding random variables (as opposed to their realizations x_t and x_1).

This derivation is crucial. It reveals that the network does not learn the individual trajectory velocity $u_t(X_t \mid X_1)$ directly. Instead, it learns $u_t^{\phi}(x, r)$: the *marginalized target velocity field conditioned on the representation manifold*. Because this true regression target $u_t^{\phi}(x, r)$ depends strictly on the partitioning of the sample space defined by f_ϕ, the parameter ϕ dictates the structural complexity of the vector field that v_θ must approximate. This mathematical reality aligns with our claim: while the ground-truth path between (x_0, x_1) is fixed, the parameter ϕ actively shapes the regression target $u_t^{\phi}(x, r)$, effectively "designing the path" from the perspective of the velocity network.

A.3. Optimization Dynamics and Representation Learning

We can further illuminate why jointly optimizing ϕ leads to high-quality discriminative representations. Using the law of total variance, we can decompose the expected loss evaluated at the optimal generator v_{θ^*}:

$$\mathcal{L}_{\text{SenFlow}}(\theta^*, \phi) = \mathbb{E}_{t,\, x_t,\, r}[\text{Var}(u_t(X_t \mid X_1) \mid X_t = x_t,\ f_\phi(X_1) = r)]\,. \tag{12}$$

The loss reduces entirely to the residual variance of the underlying velocity field within the equivalence classes defined by the encoder f_ϕ.

When we update the encoder parameters ϕ via gradient descent to minimize $\mathcal{L}_{\text{SenFlow}}$, we are explicitly minimizing this conditional variance:

$$\min_{\phi}\ \mathbb{E}\big[||u_t(X_t \mid X_1) - \mathbb{E}[u_t(X_t \mid X_1) \mid X_t,\ f_\phi(X_1)]\,||^2\big]\,. \tag{13}$$

To minimize this objective, the encoder f_ϕ is forced to extract and preserve maximal information about X_1 that explains the variations in the vector field $u_t(X_t \mid X_1)$. If the representation $r = f_\phi(X_1)$ suffers from information collapse (i.e., mapping distinct target samples X_1 to the same representation r), the residual variance increases, thereby increasing the loss. Conversely, by separating samples with divergent generative trajectories into distinct regions of the representation space, the encoder makes the vector field $u_t^{\phi}(x, r)$ smoother and easier for v_θ to approximate.

This theoretical insight explains why SenFlow achieves state-of-the-art performance on downstream Human Activity Recognition tasks, as empirically demonstrated in Section 5.2. The joint training paradigm induces a representation space that tends to separate samples with distinct temporal dynamics, which corresponds to the semantic activity classes observed empirically in Section 5.2 (Figure 4).

A.4. Summary of the Theoretical Reconciliation

In summary, the apparent contradiction is resolved by distinguishing between the *individual sample path* and the *regression target manifold*:

1. **Individual Sample Path.** The trajectory $x_t = (1-t)x_0 + tx_1$ and its velocity $u_t = x_1 - x_0$ are fixed by the linear interpolation between the source and target samples, and are structurally independent of ϕ.
2. **Regression Target Manifold.** The actual target approximated by the neural network v_θ is $u_t^{\phi}(x, r) = \mathbb{E}[x_1 - x_0 \mid X_t = x, f_\phi(X_1) = r]$. This target is dynamically parameterized by ϕ.

Therefore, our interpretation that ϕ parameterizes the target velocity field refers specifically to $u_t^{\phi}(x, r)$, the conditional expectation field. The joint optimization process mathematically corresponds to finding a representation space that makes this marginalised velocity field maximally deterministic and tractable, thereby ensuring the extraction of powerful discriminative features for self-supervised learning.

B. Robustness and Statistical Significance

To ensure the statistical reliability of our empirical claims and confirm that our performance gains are robust against random variation, we conducted fine-tuning evaluations across 5 different random seeds. Table 6 reports the mean F1-scores and standard deviations for SenFlow and the state-of-the-art baseline, SSL-Wearables. SenFlow consistently and significantly outperforms the baseline across all datasets, confirming the stability and superiority of our representation learning framework.

Table 6: HAR results with Fine-tuning. (F1-score:mean ± std)

Model	ADL	Opportunity	PAMAP2	REALWORLD	WISDM
SSL-Wearables (Yuan et al., 2024)	.829±.101	.595±.085	.789±.054	.792±.075	.810±.127
SenFlow (ours)	.929±.073	.790±.072	.883±.023	.919±.092	.888±.022

C. Extended Ablation Study on Dynamic Guidance Switching

In Section 4.1.1, we introduced Dynamic Guidance Switching (DGS) as a regularization technique that randomly masks the representation condition r. To provide deeper insights into its regularization effect, we conducted a sensitivity analysis over a wider range of masking probabilities (0%, 25%, 50%, and 75%). As shown in Table 7, the highest performance is consistently obtained around a 50% masking rate. This supports our hypothesis: moderate masking prevents the velocity field network from over-relying on the representation, while simultaneously forcing the encoder to extract only the most essential, robust features that dramatically aid generation when the condition is present.

Table 7: DGS masking ablation.

Mask prob.	ADL	Opportunity	PAMAP2	REALWORLD
0%	.861±.012	.765±.088	.872±.042	.911±.030
25%	.894±.091	.776±.083	.873±.083	.918±.042
50%	.929±.073	.790±.072	.883±.023	.919±.092
75%	.895±.019	.775±.013	.863±.108	.907±.031

D. Representation vs. Reconstruction in Time-Series Data

Recent advances in self-supervised learning for computer vision have demonstrated the power of combining Masked Image Modeling (MIM) with discriminative objectives. Representative methods include iBOT (Zhou et al., 2022), which performs online tokenization with a masked-patch prediction head, and the DINOv2 (Oquab et al., 2024) and DINOv3 (Siméoni et al., 2025) family, which scales this paradigm to large vision transformers. A related but distinct line of work, REPA (Yu et al., 2025), improves the generation quality of diffusion transformers by aligning intermediate representations with those of a *pretrained, frozen* SSL model such as DINOv2. Although REPA also connects representation learning with generative modeling, it fundamentally differs from SenFlow in that it relies on an external pretrained encoder rather than training the encoder and generator jointly from scratch. In the wearable sensor domain, powerful off-the-shelf representation models analogous to DINOv2 are not yet widely available; SenFlow is therefore designed to acquire high-quality representations and generation capabilities simultaneously, without requiring a pretrained teacher.

Directly applying the MIM paradigm to 1D continuous time-series data, such as wearable IMU signals, poses fundamental challenges due to inherent domain gaps between image and sensor modalities. Images exhibit high spatial redundancy: missing patches can be inferred from surrounding local semantics, and the reconstruction objective naturally encourages the model to learn rich, hierarchical features. High-frequency IMU signals, by contrast, are characterized by temporal continuity but lack such dense local semantic cues. When standard masking strategies are applied to sensor data, the reconstruction objective often collapses into a trivial interpolation problem: the network acts as a low-level smoothing filter and predicts masked time steps from immediately

adjacent observations. This behavior has been empirically reported in the time-series masking literature (Dong et al., 2023) and was confirmed in our own preliminary studies, where several signal-specific pretext tasks—including standard segment masking, smoothed reconstruction, temporal alignment, and spatial alignment—were explored before arriving at the present method. None of these reconstruction-based objectives proved sufficient for capturing the complex motion manifold required for downstream activity recognition.

SenFlow diverges fundamentally from masked reconstruction by employing CFM. Rather than recovering masked temporal segments, our framework requires the model to learn a continuous transformation from a structureless Gaussian prior to the complex data manifold. To synthesize an entire signal trajectory from noise, the velocity field network v_θ must rely on the conditioning representation $r = f_\phi(x_1)$ to determine the target distribution. This full-sequence generative objective strictly prevents trivial local interpolation, because no local context is available during generation. Consequently, the encoder is compelled to capture and compress the global structural dynamics and invariant kinematics of human motion into r. The DGS mechanism (Section 4.1.1) further reinforces this separation of concerns: when r is randomly masked during training, the velocity field network must maintain generative quality autonomously, which prevents the encoder from encoding redundant low-level details that can be handled by the generator alone.

We therefore argue that flow-based generative alignment provides a substantially more effective inductive bias for representation learning in time-series data than patch-level or segment-level reconstruction. This design choice is supported by the empirical results in Section 5.2, where SenFlow consistently outperforms both contrastive and reconstruction-based SSL baselines across all five evaluation datasets.