
HIGH-PERFORMANCE SELF-SUPERVISED LEARNING BY JOINT TRAINING OF FLOW MATCHING

Kosuke Ukita

Kyushu Institute of Technology
ukita.kosuke299@mail.kyutech.jp

Tsuyoshi Okita

Kyushu Institute of Technology
tsuyoshi@ai.kyutech.ac.jp

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 Foundation Model (FlowFM), 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, FlowFM accelerates and stabilizes training, improving its efficiency for representation learning. Experiments on wearable sensor data show FlowFM reduces training time by 50.4% compared to a diffusion-based approach. On downstream tasks, FlowFM surpassed the state-of-the-art SSL method (SSL-Wearables) 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/jointOptimizationFlowMatching>.

sis [25], and scientific domains such as drug discovery [12] and robotics [24].

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) [5, 33].

However, applying diffusion models to representation learning presents a fundamental dilemma. Repurposing existing generative models (e.g., DDAE [33]) yields representations not optimized for recognition tasks [34], while redesigning them for representation learning (e.g., I-DAE [5], SODA [13]) often sacrifices their powerful generative capabilities. This stems from an inherent trade-off between generative and discriminative quality [5, 7], 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 FlowFM (Flow Matching-based Foundation Model). FlowFM 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, FlowFM achieves high generative and discriminative performance without compromise, creating a more versatile and powerful foundation model.

The contributions of this paper are as follows:

- We propose FlowFM, 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

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 [36], time-series forecasting [18], speech synthe-

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.

This paper is organized as follows. In Section 2, we review related work. In Section 3, we cover the necessary background, and in Section 4, we describe our proposed method in detail. Next, in Section 5, we present the experimental results. Finally, in Section 6, we conclude the paper.

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 [6] and has since achieved tremendous success in diverse modalities, such as with Masked Autoencoders (MAE) [9] 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 [10], SimCLR [4], and DINO [3]. This paradigm is particularly known for learning features that exhibit excellent linear separability in downstream tasks.

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 [33] discovered that the intermediate representations

of diffusion models pretrained for generative tasks unintentionally contain high-quality discriminative features. DDAE [33] demonstrated a way to leverage existing models as zero-cost feature extractors. RepFusion [34] further proposed a sophisticated method that combines knowledge distillation and reinforcement learning to dynamically extract optimal features tailored to a specific task. Diffusion Classifier [14] 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. 1-DAE [5] 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 [13] 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.

2.3 Flow Matching for Generative Modeling

To address the computational cost issues of diffusion models, particularly their slow sampling speed, flow matching [15, 17] 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, potentially in a single step, without the need for iterative sampling like in diffusion models. Conditional Flow Matching [28, 29] 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 are, to the best of our knowledge, largely non-existent. This paper ventures into this unexplored territory.

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 DDPM, score-based models, and SDE-based models [11, 26], Latent Diffusion Models (LDMs) [23] construct the diffusion process in a compressed latent space, using methods like VAEs to encode pixel-space data. This approach significantly improves both the quality and speed of generation. Furthermore, the Diffusion Transformer (DiT) [20] 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).

$$x_t = (1 - t)x_0 + tx_1 \quad (1)$$

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) [16].

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

4 Proposed Method

In Section 4, we detail FlowFM, a novel foundation model that utilizes flow matching. Our method follows the conventional two-stage process of self-supervised learning [6, 3], 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 FlowFM during pre-training is shown in Figure 1. FlowFM 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).

$$r = f_\phi(x_1) \quad (3)$$

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).

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

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

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

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 [23, 1], they all use representations from a pretrained, fixed encoder, making the conditioning information static and not a target of learning. In contrast, our method is different in that it trains the Representation Encoder concurrently with the generation process. The representation in the condition changes dynamically and is updated. By training the representation via gradients from the flow matching generation process, it is expected to acquire not only generative capabilities but also advanced recognition abilities.

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

The FlowFM 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.

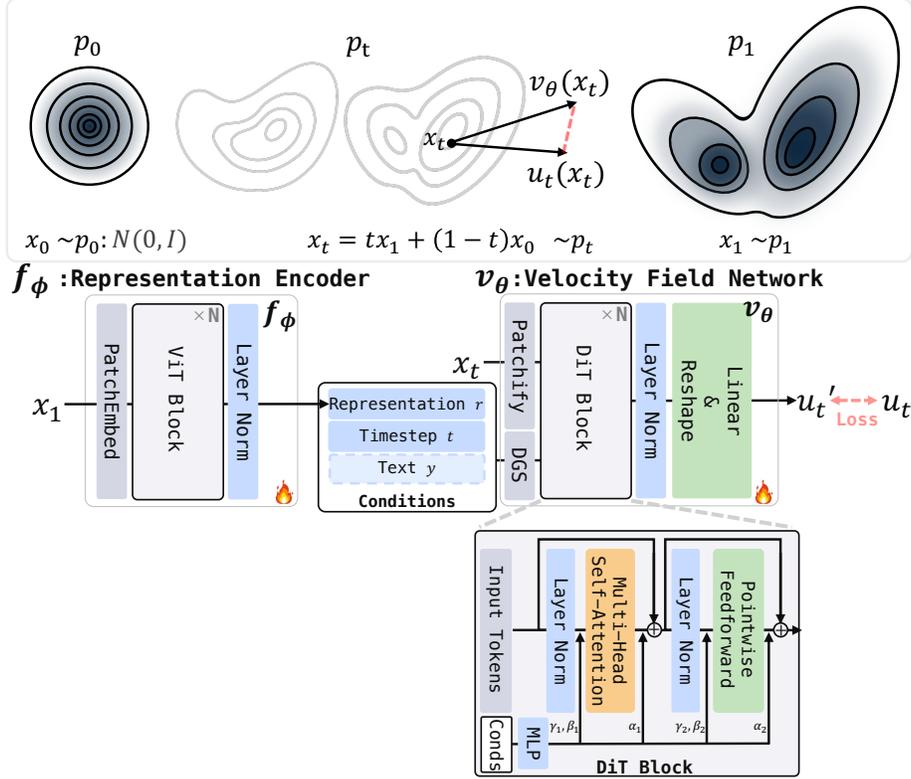


Figure 1: **Overall architecture of FlowFM during pre-training.** The model consists of (left) Representation Encoder f_ϕ and (right) Velocity Field Network v_θ . f_ϕ extracts representation r from input x_1 and supplies it as a condition to v_θ . v_θ takes point x_t on the probability path and conditions (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.

Velocity Field Network The velocity field network in this study is based on the DiT [20] 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, within the patch embedding layer at the input of DiT, we replaced it 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 Transformer-based ViT [8] architecture. We modified 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 for conditional

generation. 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 [20] 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

¹The conditioning mechanism employed in DiT.

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. Then, when the representation is provided as a condition, it must offer essential information that facilitates learning to contribute to the training of the generation process. Therefore, masking the conditioning representation is expected to be effective. 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.

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 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 FlowFM pre-training capture not just discriminative information but also richer, semantic structures, we conduct 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 FlowFM pre-training is capable of zero-shot unconditional and representation-conditioned generation, 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 initial state. Figure 2 illustrates the workflow for tuning and signal generation during inference for the Text-to-Signal

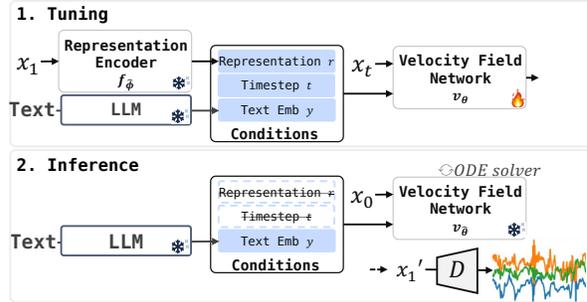


Figure 2: **Text-to-Signal workflow:** (1) In tuning, the network v_θ is fine-tuned conditioned on both representation r and text embedding y with frozen components. (2) In inference, the model generates signals x'_1 from noise x_0 via an ODE solver, conditioned solely on text y .

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 [23, 1]. 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 denoising 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	<u>9.991</u>	<u>0.6300</u>	<u>0.6597</u>
REALWORLD	Text-Only	12.33	0.2486	0.2480
	Text + Rep	<u>8.261</u>	<u>0.6838</u>	<u>0.6925</u>

Table 3: **HAR classification performance:** Comparison of FlowFM against state-of-the-art baselines (e.g., SSL-Wearables) across five datasets. FlowFM 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 [35]	-	.7540	-	.5470	-	.7250	-	.7710	-	.7230
1D-DINO	.8835	.8691	.7557	.7682	.8111	.7981	.8124	.8243	.7747	.7716
SENVt-u4 [19]	.8394	.7936	.7103	.6826	.7157	.6912	.7470	.7512	.6667	.6618
SENVt-u7 [19]	.8472	.7849	.7150	.6909	.7098	.6869	.7628	.7692	.6819	.6766
DiffFM (ours)	.9055	.8813	.7787	.7760	.8171	.8090	.8614	.8741	.8261	.8224
FlowFM (ours)	.9370	.9168	.7881	.7792	.8467	.8373	.8683	.8832	.8337	.8322
Fine-tuning										
BERT [6]	.8583	.8553	.7272	.7245	.7561	.7527	.7807	.7819	.7923	.7924
SSL-Wearables [35]	-	.8290	-	.5950	-	.7890	-	.7920	-	.8100
1D-DINO	.9024	.8780	.7656	.7741	.8341	.8223	.8878	.8978	.8693	.8684
SENVt-u4 [19]	.9087	.8780	.7489	.7411	.8488	.8454	.9015	.9124	.8693	.8679
SENVt-u7 [19]	.8992	.8684	.7429	.7448	.8477	.8445	.9058	.9163	.8875	.8867
1D-Diffusion Classifier	.8818	.8515	.7693	.7526	.8763	.8768	.8959	.8574	.8076	.8075
DiffFM (ours)	.9291	.9118	.7857	.7782	.8902	.8877	.9084	.9197	.8839	.8832
FlowFM (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.

True	0	83	1	3	0	1	0	0	0	0	0	251	8	0	1	3	1	1	3
	1	63	9	0	1	1	2	3	7	59	0	1	1	0	0	0	0	0	0
	2	7	59	0	1	1	0	0	0	0	0	0	0	46	0	0	0	0	0
	3	0	0	4	65	7	3	0	0	0	0	0	0	0	0	0	0	0	0
	4	1	0	2	9	38	3	2	0	0	0	0	0	0	0	0	0	0	0
	5	0	1	3	1	3	41	3	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	1	0	1	0	59	5	0	0	0	0	0	0	0	0	0	0
	7	0	0	1	0	0	1	3	78	0	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	

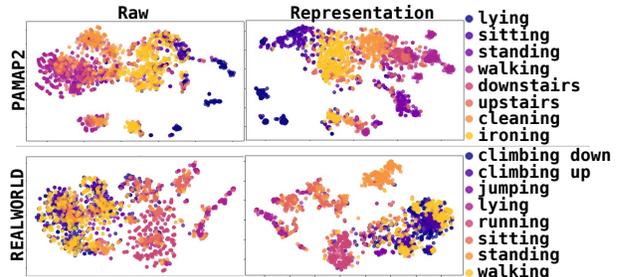


Figure 3: **Confusion matrices on HAR tasks:** The left panel displays transfer learning results on the PAMAP2 dataset, while the right panel shows fine-tuning results on the REALWORLD dataset. The strong diagonal patterns demonstrate FlowFM’s high classification accuracy across diverse activity classes.

Figure 4: **t-SNE visualization of learned representations:** Comparison between raw data (left) and FlowFM 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.

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 FlowFM. To clearly demonstrate the effectiveness of our proposed FlowFM, we designed a diffusion model-based counterpart for direct comparison. We name this model the Diffusion-based Foundation Model (DiffFM). It employs the same joint

training architecture of a representation encoder and a generative network as FlowFM, but with the generation mechanism replaced by a standard diffusion model. By directly comparing the performance and computational costs of FlowFM and DiffFM, we quantitatively evaluate the improvements in recognition performance and efficiency achieved by FlowFM.

5.1 Datasets

In this experiment, we use 3-axis accelerometer data from wrist-worn IMU sensors. Following Yuan et al. [35], we segment the signals into fixed-length sliding windows and treat them as independent inputs. All data used in the

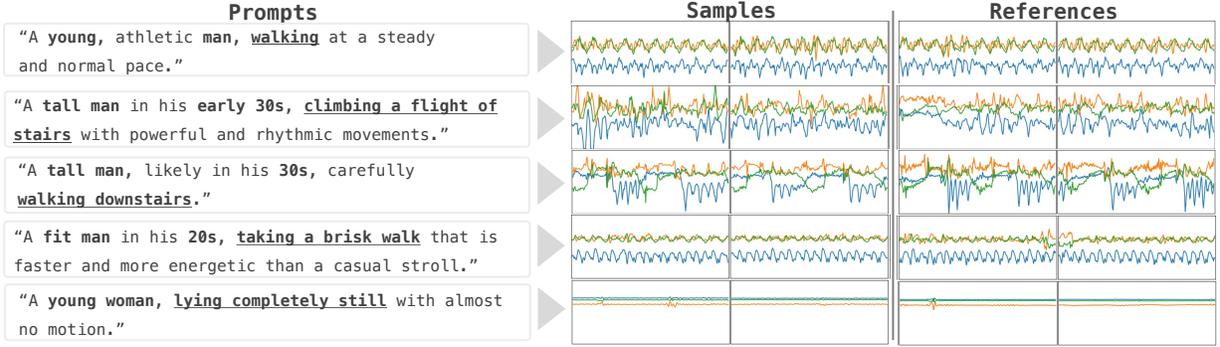


Figure 5: **Text-to-Signal generation examples:** Generated 3-axis accelerometer signals (Samples) conditioned on text prompts are shown alongside reference data (References). The model accurately translates semantic descriptions—such as activity type, intensity, and subject characteristics—into realistic time-series patterns, capturing distinct motion signatures like "walking" versus "climbing stairs."

Table 4: **Datasets**

Dataset	Subjects	Samples	Classes
Capture-24 [32]	152	1.3M	4
ADL [2]	7	0.6K	5
Opportunity [22]	4	3.9K	4
PAMAP2 [21]	8	2.9K	8
REALWORLD [27]	14	12K	8
WISDM [31]	46	28K	18

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 dataset [32], 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 [2], Opportunity [22], PAMAP2 [21], REALWORLD [27], and WISDM [31].

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 [35], a state-of-the-art multi-task SSL method for wearable data; 1D-DINO, a contrastive learning approach adapted from DINO [3]; SENvT-u4 [19] is trained using multiple pretext tasks, such as Transformer-based masking. Its variant, SENvT-u7 [19], is trained under a broader multi-task learning framework with additional tasks. 1D-Diffusion Classifier, an adaptation of Diffusion Classifier [14]; and our own DiffFM, a diffusion-based counterpart to FlowFM.

In both transfer learning and fine-tuning settings, FlowFM consistently outperformed all baselines across all five datasets. Under transfer learning, FlowFM improved upon the strong 1D-DINO baseline by up to +7.02% in accuracy and +6.06% in F1-score on REALWORLD and

WISDM. The strong performance of our diffusion-based DiffFM also validates the effectiveness of our decoupled architecture. In the fine-tuning setting, FlowFM 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.

To visualize the quality and generalization ability of the learned representations, we used t-SNE [30] 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.

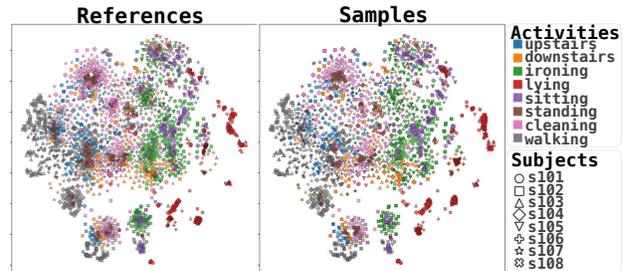


Figure 6: **t-SNE visualization of generated signals:** Comparison between real data (References) and signals generated from text conditions (Samples) on the PAMAP2 dataset. The generated distribution closely mirrors the real data, demonstrating that the model successfully captures distinct clusters for both activity types (colors) and individual subject characteristics (shapes).

5.3 Text-to-Signal Results

To verify that the learned representations capture rich semantic structures beyond discriminative features, we conducted 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 LLM Llama3 (8B) in a modular text encoder designed for future extensibility. As shown in Figure 5, 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 6) shows that the distribution of text-generated signals closely mirrors that of the original data, successfully distinguishing between both activities and subjects.

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

FlowFM 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 models. 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), FlowFM (3.2h) succeeded in reducing the training time by 50.4% compared to DiffFM (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.

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

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

In the inference phase, FlowFM showed a distinct advantage in generation speed. As shown in Table 5, FlowFM achieved up to a 51.0x speedup over DiffFM while maintaining high generation quality (FID: 5.014). This speedup is due to the different generative processes: FlowFM 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 (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. This speed improvement highlights its potential contribution to real-time processing and edge AI applications.

6 Conclusion

In this work, we proposed FlowFM, a novel foundation model that applies the generative capabilities of flow matching to representation learning. FlowFM 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, FlowFM outperformed all existing SSL methods, including the state-of-the-art SSL-Wearables, across all five datasets. This result quantitatively demonstrates that FlowFM, based on its new principles, can acquire versatile representations applicable to a diverse range of downstream tasks. Furthermore, in terms of computational efficiency, FlowFM significantly reduced training time compared to the diffusion-based DiffFM, halving the total energy consumption. In the Text-to-Signal generation task, it achieved up to a 51.0x speedup over DiffFM while maintaining high generative quality, demonstrating that energy consumption can be dramatically reduced without sacrificing generative quality.

Based on these results, we conclude that FlowFM 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 quality. These characteristics position FlowFM as a vital step towards realizing on-device foundation models that operate within strict latency and energy budgets. Ultimately, our approach provides a scalable blueprint for sustainable AI, ensuring that the benefits of large-scale representation learning are accessible even in resource-limited edge environments. Future work includes applying this framework to diverse modalities such as images and audio and adapting it to more complex downstream tasks.

References

- [1] Florian Bordes, Randall Balestriero, and Pascal Vincent. High fidelity visualization of what your self-supervised representation knows about. *arXiv preprint arXiv:2112.09164*, 2021.
- [2] Barbara Bruno, Fulvio Mastrogiovanni, Antonio Sgorbissa, Tullio Vernazza, and Renato Zaccaria. Analysis of human behavior recognition algorithms based on acceleration data. pages 1602–1607, 2013.
- [3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [5] Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. *arXiv preprint arXiv:2401.14404*, 2024.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019.
- [7] Mischa Dombrowski, Hadrien Reynaud, Johanna P Müller, Matthew Baugh, and Bernhard Kainz. 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, 2024.
- [8] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.
- [10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.
- [11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [12] Lei Huang, Tingyang Xu, Yang Yu, Peilin Zhao, Xingjian Chen, Jing Han, Zhi Xie, Hailong Li, Wenge Zhong, Ka-Chun Wong, et al. A dual diffusion model enables 3d molecule generation and lead optimization based on target pockets. *Nature Communications*, 15(1):2657, 2024.
- [13] Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. Soda: Bottleneck diffusion models for representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23115–23127, 2024.
- [14] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. *arXiv preprint arXiv:2303.16203*, 2023.
- [15] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- [16] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.
- [17] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.
- [18] Caspar Meijer and Lydia Y Chen. The rise of diffusion models in time-series forecasting. *arXiv preprint arXiv:2401.03006*, 2024.
- [19] Tsuyoshi Okita, Kosuke Ukita, Koki Matsuishi, Masaharu Kagiya, Kodai Hirata, and Asahi Miyazaki. 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*, pages 499–504, 2023.
- [20] William Peebles and Saining Xie. Scalable diffusion models with transformers. pages 4195–4205, 2023.
- [21] Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. *2012 16th international symposium on wearable computers*, pages 108–109, 2012.
- [22] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczeck, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. Collecting complex activity datasets in highly rich networked sensor environments. *2010 Seventh international conference on networked sensing systems (INSS)*, pages 233–240, 2010.

- [23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [24] Yorai Shaoul, Itamar Mishani, Shivam Vats, Jiaoyang Li, and Maxim Likhachev. Multi-robot motion planning with diffusion models. *arXiv preprint arXiv:2410.03072*, 2024.
- [25] Kai Shen, Zeqian Ju, Xu Tan, Yanqing Liu, Yichong Leng, Lei He, Tao Qin, Sheng Zhao, and Jiang Bian. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *arXiv preprint arXiv:2304.09116*, 2023.
- [26] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [27] Timo Sztyler and Heiner Stuckenschmidt. On-body localization of wearable devices: An investigation of position-aware activity recognition. pages 1–9, 2016.
- [28] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.
- [29] Alexander Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. *arXiv preprint arXiv:2307.03672*, 2023.
- [30] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [31] Gary M Weiss, Kenichi Yoneda, and Thaier Haya-jneh. Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access*, 7:133190–133202, 2019.
- [32] Matthew Willetts, Sven Hollowell, Louis Aslett, Chris Holmes, and Aiden Doherty. Statistical machine learning of sleep and physical activity phenotypes from sensor data in 96,220 uk biobank participants. *Scientific reports*, 8(1):1–10, 2018.
- [33] Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15802–15812, 2023.
- [34] Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18938–18949, 2023.
- [35] Hang Yuan, Shing Chan, Andrew P Creagh, Catherine Tong, Aidan Acquah, David A Clifton, and Aiden Doherty. Self-supervised learning for human activity recognition using 700,000 person-days of wearable data. *npj Digital Medicine*, 7(1):1–18, 2024.
- [36] Haiyu Zhang, Xinyuan Chen, Yaohui Wang, Xi-hui Liu, Yunhong Wang, and Yu Qiao. 4diffusion: Multi-view video diffusion model for 4d generation. *Advances in Neural Information Processing Systems*, 37:15272–15295, 2024.