2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

VANP: Learning Where to See for Navigation with Self-Supervised Vision-Action Pre-Training

Mohammad Nazeri, Junzhe Wang, Amirreza Payandeh, and Xuesu Xiao

***Abstract*— Humans excel at efficiently navigating through crowds without collision by focusing on specific visual regions relevant to navigation. However, most robotic visual navigation methods rely on deep learning models pre-trained on vision tasks, which prioritize salient objects—not necessarily relevant to navigation and potentially misleading. Alternative approaches train specialized navigation models from scratch, requiring significant computation. On the other hand, self-supervised learning has revolutionized computer vision and natural language processing, but its application to robotic navigation remains underexplored due to the difficulty of defining effective self-supervision signals. Motivated by these observations, in this work, we propose a Self-Supervised Vision-Action Model for Visual Navigation Pre-Training (VANP). Instead of detecting salient objects that are beneficial for tasks such as classification or detection, VANP learns to focus only on specific visual regions that are relevant to the navigation task. To achieve this, VANP uses a history of visual observations, future actions, and a goal image for self-supervision, and embeds them using two small Transformer Encoders. Then, VANP maximizes the information between the embeddings by using a mutual information maximization objective function. We demonstrate that most VANP-extracted features match with human navigation intuition. VANP achieves comparable performance as models learned end-to-end with half the training time and models trained on a large-scale, fully supervised dataset, i.e., ImageNet, with only 0.08% data.** [1]

Fig. 1: **Comparison of Activation Maps Learned by End-to-End, ImageNet, and VANP.** VANP can extract multiple regions of interest for navigation without downstream navigation supervision compared to single salient regions by End-to-End and ImageNet pre-trained models.

I. Introduction

In recent years, imitation learning, particularly behavior cloning [1], has become a leading approach for visual navigation models [2]–[4]. However, the performance of these models heavily relies on the visual features extracted by the model's visual encoder. Although the limited memory and processing power onboard robots restrict the size of models deployable in real time, with such limitations we still need accurate and efficient onboard visual encoders, making convolutional neural networks (CNNs) more desirable than larger Vision Transformer models (ViTs) [5].

Training a visual navigation-specific encoder from scratch requires a large amount of data, leading to high computational demands and extended training times [6]. To reduce this computational burden, most approaches use pre-trained vision models [3]. While these models provide a decent scene representation, they specialize in extracting salient features for vision tasks such as object classification and detection [7]. These features may not always align with what is crucial for navigation [8]. For example, following sidewalks, avoiding grass, or navigating around stairs and guardrails are essential for robots, but these features might not be captured by encoders trained for generic vision tasks. Consequently, pre-trained models, like those trained on ImageNet, can sometimes lead to navigation failures by focusing on irrelevant distractions.

Self-Supervised Learning (SSL) has shown success in various computer vision tasks by extracting general features adaptable to downstream tasks with/without fine-tuning. However, a discrepancy exists between features extracted from generic models and those specifically needed for navigation. This leads us to ask the question: *can we train visual encoders that extract only navigation-relevant features using self-supervision?*

Considering both the success of SSL on a variety of computer vision tasks and the oftentimes mismatched features provided by generic SSL models for navigation tasks, we present Vision-Action Navigation Pretraining (VANP), a non-contrastive self-supervised approach that completely

All authors are with the Department of Computer Science, George Mason University {mnazerir, jwang69, apayande, xiao}@gmu.edu. This work has taken place in the RobotiXX Laboratory at George Mason University. RobotiXX research is supported by National Science Foundation (NSF, 2350352), Army Research Office (ARO, W911NF2220242, W911NF2320004, W911NF2420027), US Air Forces Central (AFCENT), Google DeepMind (GDM), Clearpath Robotics, and Raytheon Technologies (RTX).

[1]Full version: https://arxiv.org/abs/2403.08109

relies on a navigation-specific pretext task to train the visual encoder without the need for negative samples.

The core idea behind VANP is inspired by how humans navigate in crowded spaces. We do not need to pay attention to all the people and objects in the scene, but only the ones that affect our navigation trajectory. To this end, VANP embeds visual history, future actions, and visual goal as self-supervision signals and leverages Transformers with additional context tokens (inspired by Bert [9] and Vision-Transformers [5]) to generate embeddings. Then, VANP utilizes VICReg [10] as the pretext objective function to maximize the mutual information between the embeddings. The trained visual encoder can therefore discard redundant features unnecessary for navigation and focus only on navigation-relevant regions. For example, Fig. 1 shows the activation map of the last layer of ResNet-50 [11] trained with different methods. VANP learns navigation-relevant visual features with the help of our navigation-specific self-supervision signals.

Our experimental results suggest that VANP-extracted features trained on a dataset [12] that only contains 0.08% samples compared to ImageNet are as informative for a downstream navigation task as using ImageNet features. The contributions of this work can be summarized as follows:

- An SSL framework to train a visual encoder for robotic navigation tasks;
- Insights into what is happening inside CNNs during navigation using different approaches; and
- A benchmark on short and long-term navigation interaction to show the performance of different approaches.

II. Related Work

Recent advances in natural language processing and computer vision, particularly those driven by self-supervised learning (SSL), motivate our work. In this section, we compare SSL approaches for representation learning.

Codevilla *et al.* [3] demonstrated the value of pre-trained models for training better policies in autonomous vehicles. Subsequently, many works adopted pre-trained computer vision models, often trained on ImageNet [3], [4]. However, general-purpose "foundation models" pre-trained on pretext tasks can achieve richer representations, enabling them to generalize to various downstream tasks with minimal data in a zero- or few-shot manner.

The literature has extensively studied foundation models for robot manipulation [13], [14]. For example, R3M [15] pre-trained a general visual encoder for manipulation tasks. Dadashi *et al.* proposed AQuaDem [13], a framework to learn quantized actions from demonstrations in continuous action spaces, while VANP is doing the opposite by learning visual features from continuous action spaces. Luo *et al.* [14] improved AQuaDem by using VQ-VAE [16] for offline reinforcement learning.

Shen *et al.* proposed conditioning visual demonstrations like segmentation and depth maps on actions during fusion rather than employing a naive fusion approach [17]. Yang *et al.* [18] projected the visual cues for navigation on the image space and then trained a policy on the augmented image. STERLING [19] and CAHSOR [20] have explored the concept of human preference learning and competence-awareness in the context of off-road navigation using SSL. These methods aligned sensor and visual embeddings by maximizing the mutual information between embeddings by leveraging VICReg [10] and BarlowTwins [21] respectively.

The work by Eftekhar *et al.* [22] presented the closest approach to VANP, employing a learnable codebook module to selectively filter visual observations based on the specific task. However, relying on task-relevant information, e.g., picking up the key, requires additional information that is not available without human annotation or using a simulator while VANP does not need access to such information to learn visual features. Another work closely related to VANP is NavFormer [23], which utilized BYOL [24] on two input images retrieved from a simulator. These images differ in the presence of dynamic objects within the scene. However, this approach confines NavFormer to the simulated environment, limiting its applicability to simulation environments where we have full control of the environment, e.g., making objects invisible to learn the importance of the presence and absence of the object as an obstacle. Conversely, VANP achieves real-world data generalization without relying on the pre-definition of specific rules only possible in simulation or through human annotation.

III. Methodology

We formally define the visual navigation task and the learning setting for VANP.

A. Problem Definition

We define visual navigation as the task of navigating an environment with only RGB camera input, as explored in previous works [2]–[4]. The visual navigation problem can be formalized as follows. **Input:** The robot is given a sequence of past and current images from its front-facing camera, $o_t = [I_{t-\tau_P}, I_{t-\tau_P+1}, \ldots, I_t] \in \mathcal{O}$, where t is the current time step, τ_P is the number of past frames, and $\mathcal{O}$ is the space of all possible image sequences. The robot is also given its current goal e.g., GPS coordinates, pose, image, or next local coordinate in 2D space, $g \in \mathcal{G}$, which determines the direction it should move in the next time step. **Output:** The robot must select an action $a_t \in \mathcal{A}$ consisting of continuous linear and angular velocities. $\mathcal{A} = [-1, 1]^2$ is the action space, where $[-1, 1]$ maps to the minimal and maximal linear and angular velocity of the robot. **Visual Navigation:** The goal is to learn a policy, $\pi_\theta : \mathcal{O} \times \mathcal{G} \to \mathcal{A}$, where θ represents the policy's parameters, to determine which action to take at each time step to reach its goal efficiently while avoiding collisions with others.

End-To-End models: For end-to-end or holistic models, we define the policy π_θ as follows: $a = \pi_\theta(o, g) = \sigma_\zeta(p_\phi(o) \oplus q_\psi(g))$, where σ is the controller policy parametrized by ζ, p is the image encoder parameterized by ϕ, q is the goal encoder parameterized by ψ, and $\oplus$ is the aggregation of two vectors. To learn these parameters, two

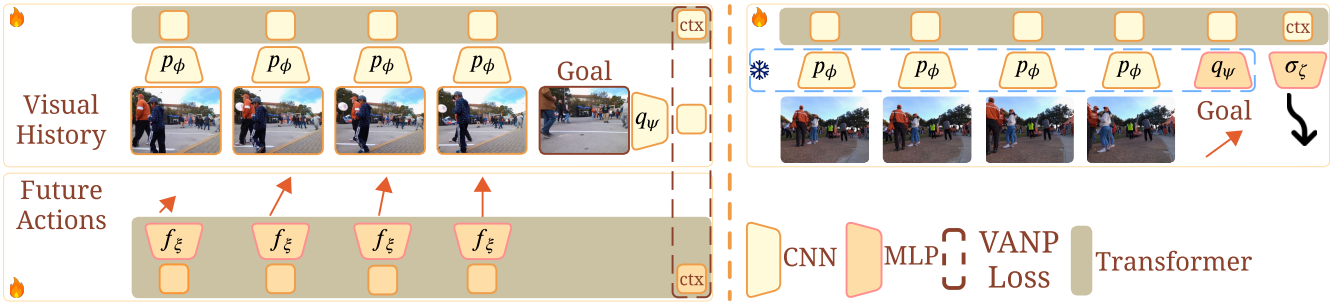

Fig. 2: **VANP Architecture.** VANP learns to embed temporal features into spatial features by using a sequence of images and leveraging two TransformerEncoders with context tokens. VANP's loss maximizes the mutual information between history, future actions, and the goal (left). Then, by appending an MLP to the Transformer context token, VANP predicts future trajectories during the downstream navigation task (right).

common approaches are (1) to learn all of them together in an end-to-end manner which makes the training difficult and time-consuming or (2) to pre-train the image encoder separately and only fine-tune the goal encoder along with the controller to reduce training time.

B. Vision-Action Model

VANP leverages VICReg [10] to maximize the information between past observations, a future goal, and future actions while maintaining the information collapse between input heads to train the image encoder p. Unlike vision SSL models that work on the joint embedding of augmented images, VANP correlates the action space $\mathcal{A}$ and goal space $\mathcal{G}$ with the pixel latent space $\mathcal{O}$ as shown in Fig. 2. We define VANP pre-training as follows: We sample a batch of $(I^i_{t-\tau_P:t}, a^i_{t:t+\tau_F}, g^i_t)$ from dataset $\mathcal{D}$, where i is the sample number, $I^i_{t-\tau_P:t}$ is a sequence of past visual observations starting from $t-\tau_P$ and ending at t, $a^i_{t:t+\tau_F}$ is a sequence of future actions starting from t and ending at $t+\tau_F$, and g^i_t is the current goal at time t instantiated as an image in the future $I^i_{t+\tau_F}$. τ_F is the number of frames in the future and τ_P is the number of frames in the past. We then feed $I^i_{t-\tau_P:t}$ to p_ϕ, typically a CNN, and all the embeddings to a transformer encoder [25], as well as $a^i_{t:t+\tau_F}$ to f_ξ as part of another transformer encoder, to learn image Z^i and action Z^a embeddings, respectively. Each transformer contains an additional context token to capture the continuous information among frames. We feed g^i_t to p_ϕ to generate goal embedding Z^g. Finally, we use VANP's objective function to learn ϕ and ξ:

$$\begin{aligned}\mathcal{L}_{\text{VANP}}(Z^i, Z^g, Z^a) = \lambda \mathcal{L}_{\text{VICReg}}(Z^i, Z^g) \\ + (1-\lambda)\mathcal{L}_{\text{VICReg}}(Z^i, Z^a),\end{aligned} \tag{1}$$

where λ is the importance of each term, and $\mathcal{L}_{\text{VICReg}}$ is the VICReg objective function [10] defined as:

$$\begin{aligned}\mathcal{L}_{\text{VICReg}}(Z^1, Z^2) = \mu^1 s(Z^1, Z^2) \\ + \mu^2[v(Z^1) + v(Z^2)] \\ + \mu^3[c(Z^1) + c(Z^2)].\end{aligned} \tag{2}$$

s is the distance between embedding spaces, v and c are the variance and covariance of each embedding respectively. μ^1, μ^2, and μ^3 are hyper-parameters controlling the effectiveness of each term. Leveraging VICReg's objective function offers the advantage of circumventing the need for negative samples, which, as mentioned above, is challenging to define within the action space for navigation tasks.

C. Implementation Details

We implement VANP with PyTorch and the training is performed on a single A5000 GPU with 24GB memory[2].

Model architecture: Considering the limited computation resources onboard most mobile robots, we choose ResNet-50 [11] without the classification head as a low-latency image encoder for p_ϕ and we call it VANP-50. We use two TransformerEncoders with additional context vectors [5], [9] with four layers and four heads as the final image and action encoders to produce the embeddings of $Z^i, Z^a \in \mathbb{R}^{512}$. Both encoders are followed by MLPs with three layers as the projection heads to generate the final $Z'^i, Z'^a \in \mathbb{R}^{1024}$. We apply the same p_ϕ to the goal image to generate $Z^g \in \mathbb{R}^{512}$. A critical challenge arises from the inherent differences in modalities between the two networks generating the embeddings, leading to significant variations in their output ranges. To address this discrepancy and ensure effective integration, we initialize all deep networks using the Kaiming Normal initialization [11] with a mean of zero and a variance of one. In the context of the downstream model, an MLP is appended to the Transformer's context vector to predict trajectories at three and five seconds into the future, enabling the evaluation of how the extracted features influence both short-term and long-term interactions.

Optimization: We use the ADAMW optimizer [26] and train the model for 200 epochs with a batch size of 2048 and a learning rate of $5e^{-4}$. We observe that large batch sizes add more variation to the update stage and improve learning. To ensure a fair comparison, all models are trained for 50 epochs using the same optimizer and hyperparameters during downstream training. The sole exception is the end-to-end model, which requires 100 epochs to guarantee convergence.

[2] https://github.com/mhnazeri/VANP

TABLE I: **Downstream Performance.** Comparison of the performance of the visual encoders with different pre-training methods on unseen data. Models denoted by an ⧗ require double the training time compared to models with ⧗ .

Type		Method	Weight	Single-frame	Multiple-frame	Frozen ❄		Fine-tuned 🔥	
						3s	5s	3s	5s
End-to-End	⧗	Resnet-50	Random	✓	✗	-	-	0.116	0.307
		ResnetTransformer	Random	✗	✓	-	-	0.113	0.320
Backbone Supervised	⧗	Resnet-50	ImageNet	✓	✗	**0.129**	0.356	0.129	0.342
		ResnetTransformer	ImageNet	✗	✓	0.169	0.435	0.107	0.292
Backbone Self-Supervised	⧗	Resnet-50	VANP	✓	✗	0.144	0.374	**0.103**	**0.272**
		ResnetTransformer	VANP	✗	✓	0.133	**0.342**	0.114	0.319

Fig. 3: **Qualitative Comparison.** Comparison of the last layer activation maps among different methods on unseen scenarios.

Dataset: We leverage a selection of two unique datasets: SCAND [12] and MuSoHu [27], both of which encapsulate robot and human navigation data from the egocentric perspective. A fundamental limitation of SSL models is their susceptibility to data quality. As we will discuss in the limitations section (Sec. IV-D), VANP is similarly affected, particularly in scenarios where there is no change in a sequence of images as shown in Fig. 4. To minimize data ambiguity and noise, a subset of the two datasets are carefully curated, ensuring representation of both indoor and outdoor scenes. The resulting dataset, comprising approximately 11,000 samples, is used for both pre-training and training phases. Additionally, a separate set of 8,000 unseen samples are used for downstream navigation task evaluation. For pretext task training, we set τ_P and τ_F to 6 and 20 respectively and use a sequence of images $I_{t-\tau_P+1:t} \in \mathbb{R}^{\tau_P \times 98 \times 126}$ along with a goal image $g_t \in \mathbb{R}^{98 \times 126}$ and a sequence of actions $a_{t:t+\tau_F-1} \in \mathbb{R}^{\tau_F \times 2}$ parsed at 4 Hz, comprising of 1.5 seconds in the past and 5 seconds in the future. For the downstream task, we use a sequence of past observations $I_{t-\tau_P+1:t} \in \mathbb{R}^{\tau_P \times 98 \times 126}$ along with the polar coordinates of the next local goal $g \in \mathbb{R}^2$ parsed at 4 Hz, containing 1.5 seconds history as the network input to produce the actions $\mathcal{A}_{t:t+\tau_F-1} \in \mathbb{R}^{\tau_F \times 2}$ for three and five seconds in the future.

IV. Experimental Results

We provide experimental results using VANP compared against a ResNet-50 pre-trained on ImageNet and end-to-end from scratch as baselines.

A. Results Discussion

We assess the efficacy of VANP pretext training by quantitatively comparing its performance with that of a ResNet-50 model [11] pre-trained on the ImageNet ILSVRC-2012 dataset [7]. This serves as the baseline alongside another ResNet-50 model trained end-to-end with randomly initialized weights. To guarantee a fair comparison, the architectures of all other components within the downstream task remain unchanged. Table I presents the mean squared error between the predicted and ground truth trajectories for short- (three seconds) and long-term (five seconds) interactions under two conditions. In the first condition, only the goal encoder and controller are trained during the downstream navigation task, while the image encoder weights are frozen. In the second condition, we compare the performance by unfreezing the image encoder weights to enable fine-tuning.

The results in Table I demonstrate that VANP achieves comparable performance to the end-to-end trained model while requiring only half the training time. Furthermore, VANP pre-trained model achieves comparable performance to ImageNet model with only 0.08% of the data size required by ImageNet, highlighting how informative the extracted representations are for navigation.

When provided with a sequence of past observations, VANP exhibits a superior ability (0.342) to utilize this additional data compared to ImageNet model when frozen (0.435). Although the ImageNet weights appear unable to leverage the temporal features provided by the transformer component when freezing its weights (Table I, row four compared against row three), fine-tuning the ImageNet model leads to performance improvement from 0.435 to 0.292, suggesting that it can better capture underlying temporal features provided by the Transformer through fine-tuning.

However, we do not see such an improvement in the case of VANP. The negligible improvement in accuracy from 0.342 to 0.319 for VANP during fine-tuning can be attributed to two reasons. First, the focus on multiple navigation-related visual regions of VANP's pre-trained weights (Fig. 3 last row) impedes adaptation/forgetting during fine-tuning compared to the ImageNet weights. Second, the temporal features from the Transformer are already in VANP weights and therefore does not require much fine-tuning. Overall, it is likely that forgetting/updating weights can be easier when the visual encoder is trained using only one single scalar training

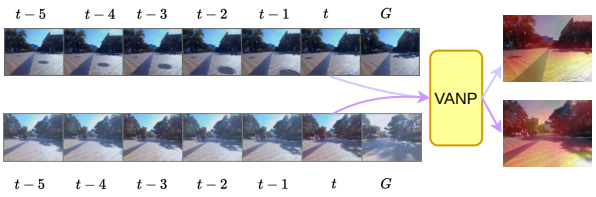

Fig. 4: **Failure Cases.** Samples without any important intra-frame changes cause the model to collapse.

TABLE II: **Ablations.** Ablation study on the role of each module on the downstream navigation task performance.

Information	3s	5s
Actions	0.167	0.499
Goal	0.160	0.392
Actions+GoalIn	0.155	0.386
Actions+GoalOut	0.144	0.383
Augmentations	**0.133**	**0.342**

loss rather than pre-trained on richer instructive signals, i.e., VANP's pre-training objective signal.

Visual inspection of the learned activation maps on the last layer of ResNet-50 (Fig. 3) reveals distinct characteristics across the models. The last row on Fig. 3 shows that the VANP pre-trained model exhibits activation maps with a higher degree of relevance to navigation tasks, focusing on features such as paths and obstacles while the ImageNet pre-trained model (Fig. 3 third row) primarily focuses on salient objects within the environment, which might not be directly related to navigation. Another difference between VANP and the end-to-end model (Fig. 3 second row) is that the end-to-end model tends to concentrate on a single critical region significantly impacting the trajectory, likely due to its limited instructive signal during training, i.e., minimizing the distance between predicted and ground truth trajectories. Conversely, VANP demonstrates the ability to extract information from multiple regions, potentially benefiting from the richer information provided by the goal image and future actions during the pre-training stage. However, as mentioned above, this richness impedes adaptation during fine-tuning.

We observe instances where the attention of all models shifts to seemingly irrelevant aspects. In the case of VANP, we posit that this may be due to the robot's sharp turns temporarily obscuring the goal image from the current frame.

B. Ablations

To investigate the most effective approach for correlating visual and action spaces, we conduct a series of ablation studies, in which we report the performance on the downstream navigation task in Table II.

Role of Different Training Signals: We assessed the individual contributions of various self-supervised training signals by changing the value of λ between 0 and 1 in Eq. 1. Our findings reveal that while action signals provide valuable navigational cues, their sparsity often hinders their effectiveness in downstream navigation tasks, especially during long-term interactions. Conversely, information derived from the goal, while occasionally exhibiting redundancy, improved performance from 0.499 to 0.392 during long-term interactions over using only actions due to informative cues alongside the redundant elements. However, this redundancy poses challenges for the policy network, which can be remedied by more training epochs and a deeper policy network. By combining these two embeddings as the self-supervision signal, the final model can effectively learn informative features while mitigating the impact of redundant information within the embedding.

Leveraging Goal Information: We further investigated the optimal utilization of future goal information. Our findings suggest that employing the goal solely as a supervision signal (shown as Actions+GoalOut in Table II) proves more effective in facilitating the model's learning of visual features compared to incorporating the goal directly within the Transformer architecture (shown as Actions+GoalIn in Table II). The Transformer's ability to capture temporal changes from the current to the goal frame is only helpful when the goal is visible from the current frame.

Augmentations: Data augmentation is a standard technique employed to enhance model generalization by introducing variability into the dataset. We follow the augmentation scheme outlined by Bardes *et al.* [10] and the result is shown as Augmentations in Table II. We observe that random cropping is particularly critical for VANP, especially in scenarios exemplified by Fig. 4, as it introduces inter-frame variation. This augmentation strategy relaxes the assumption of carefully curated data and enables an expansion of the dataset from 11,000 to 26,042 samples to include even ambiguous and noisy samples with a little performance hit.

C. Robot Deployment

To demonstrate the practical applicability of the learned visual features for navigation, a proof-of-concept demonstration of VANP-18 with a moving goal objective is deployed on a Clearpath Jackal robot. The obstacle avoidance capabilities of VANP are evaluated under controlled conditions. In these experiments, a static obstacle is initially positioned in the robot's path. Results indicate that VANP exhibits an ability to detect and avoid both static and dynamic obstructions in the majority of test cases. The supplementary video provides a record of these experiments[3]. Despite VANP's intended versatility across diverse conditions, inherent limitations considering safety only allow it to work in uncluttered environments, as elaborated in the subsequent section.

D. Limitations

We identify multiple key limitations of the VANP pre-training approach. First, our analysis of the learned kernels suggests that VANP performs more effectively when the goal image is directly visible from the current image, likely due to its reliance on image correlation for learning. While this

[3]https://youtu.be/SEuD9hkwXxQ

is helpful for Visual-Goal navigation task, it highlights a potential limitation in generalizability to scenarios where the goal location may not be directly visible from the starting point. Second, in large-scale datasets likely with a significant amount of noise, scaling VANP poses a potential challenge, considering its need for high-quality self-supervision during pre-training can result in many changes in learned activation maps between epochs. As can be seen in Fig. 4, the VANP objective is unable to learn from scenarios where there is no intra-frame change as the time passes. This limitation can be alleviated with augmentations, particularly random cropping, but it does not eliminate it. Additionally, our current findings are based on a static dataset and may not directly translate to challenging real-world navigation tasks that involve dynamic environments and unforeseen obstacles. Further research is needed to evaluate VANP's performance in these more complex scenarios.

V. Conclusions and Future Work

In this work, we propose a self-supervised learning approach to train visual encoder models specifically designed for visual navigation. This approach is motivated by the observation that humans only pay attention to specific navigation-relevant regions of their frontal view to efficiently make navigation decisions. By reversing this observation, we use the navigation decisions to extract only visual features that are relevant to the navigation task, unlike computer vision models that mainly extract salient details, which are potentially irrelevant to navigation tasks and can therefore lead to confusion for neural-based controllers. To achieve this, we leverage two Transformer Encoders to embed past visual observation, future actions, and a goal image, then we maximize the information between these embeddings using VANP's objective function to learn visual backbone weights.

Furthermore, the VANP objective function facilitates the integration of additional embeddings derived from diverse modalities, including depth data and semantic information or inputs from other sensors such as LiDARs. Studying the effectiveness of this enrichment of the embedding space with supplementary information for downstream navigation tasks can be a potential future work. Another future direction is to merge datasets from different environments, such as indoor, outdoor, off-road, and social environments, to extend the generalizability of the proposed VANP approach. More real-world experiments can support all these future directions and scale up the model to larger datasets.

References

[1] D. A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 1. Morgan-Kaufmann, 1988.

[2] F. Codevilla, M. Muller, A. Lopez, V. Koltun, and A. Dosovitskiy, "End-to-End Driving Via Conditional Imitation Learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, May 2018, pp. 4693–4700.

[3] F. Codevilla, E. Santana, A. M. Lopez, and A. Gaidon, "Exploring the Limitations of Behavior Cloning for Autonomous Driving," in *The IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2019.

[4] M. H. Nazeri and M. Bohlouli, "Exploring Reflective Limitation of Behavior Cloning in Autonomous Vehicles," in *2021 IEEE International Conference on Data Mining (ICDM)*. Auckland, New Zealand: IEEE, Dec. 2021, pp. 1252–1257.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Jun. 2021.

[6] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "ViNT: A foundation model for visual navigation," in *7th Annual Conference on Robot Learning*, 2023.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009.

[8] K. Vishniakov, Z. Shen, and Z. Liu, "ConvNet vs Transformer, Supervised vs CLIP: Beyond ImageNet Accuracy," Jan. 2024.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. 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*, Jun. 2019.

[10] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-invariance-covariance regularization for self-supervised learning," in *International Conference on Learning Representations*, 2022.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[12] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, "Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, 2022.

[13] R. Dadashi, L. Hussenot, D. Vincent, S. Girgin, A. Raichuk, M. Geist, and O. Pietquin, "Continuous control with action quantization from demonstrations." PMLR, 2022.

[14] J. Luo, P. Dong, Y. Zhai, Y. Ma, and S. Levine, "RLIF: Interactive Imitation Learning as Reinforcement Learning," Nov. 2023.

[15] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3M: A universal visual representation for robot manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 892–909.

[16] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," *arXiv preprint arXiv:1711.00937*, 2017.

[17] W. Shen, D. Xu, Y. Zhu, L. Fei-Fei, L. Guibas, and S. Savarese, "Situational Fusion of Visual Representation for Visual Navigation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 2881–2890.

[18] H.-K. Yang, T.-C. Chiang, T.-R. Liu, C.-W. Huang, J.-M. Liu, and C.-Y. Lee, "Virtual Guidance as a Mid-level Representation for Navigation," Sep. 2023.

[19] H. Karnan, E. Yang, D. Farkash, G. Warnell, J. Biswas, and P. Stone, "STERLING: Self-Supervised Terrain Representation Learning from Unconstrained Robot Experience," Oct. 2023.

[20] A. Pokhrel, A. Datar, M. Nazeri, and X. Xiao, "CAHSOR: Competence-Aware High-Speed Off-Road Ground Navigation in SE(3)," *arXiv preprint arXiv:2402.07065*, 2024.

[21] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow Twins: Self-Supervised Learning via Redundancy Reduction," Jun. 2021.

[22] A. Eftekhar, K.-H. Zeng, J. Duan, A. Farhadi, A. Kembhavi, and R. Krishna, "Selective Visual Representations Improve Convergence and Generalization for Embodied AI," Nov. 2023.

[23] H. Wang, A. H. Tan, and G. Nejat, "NavFormer: A Transformer Architecture for Robot Target-Driven Navigation in Unknown and Dynamic Environments," *arXiv preprint arXiv:2402.06838*, 2024.

[24] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[26] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," Jan. 2019.

[27] D. M. Nguyen, M. Nazeri, A. Payandeh, A. Datar, and X. Xiao, "Toward human-like social robot navigation: A large-scale, multi-modal, social human navigation dataset," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023.