JEP-VGP: LEARNING VISION GRAPH PROMPTS VIA JOINT-EMBEDDING PREDICTION OF PROMPT RESIDUALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision Graph Neural Networks (ViGs) model irregular semantic relationships in images through dynamic visual graphs, while vision graph prompting enables parameter-efficient adaptation by introducing low-rank Graph, Edge, and Node Prompts with a frozen backbone. However, existing methods mainly rely on image-level classification supervision, providing limited contextual constraints on prompt-induced node representation changes and underutilizing cross-region dependencies in visual graphs. To address this issue, we propose JEP-VGP, a vision graph prompt learning framework inspired by the Joint-Embedding Predictive Architecture (JEPA), which takes prompt residuals as prediction targets to provide explicit contextual guidance. Specifically, we introduce Prompt-Residual Embedding Prediction (PREP), which extracts prompt-induced representation changes by comparing prompted and matched-reference representations under the same configuration, and predicts target residuals from contextual residuals through a lightweight predictor with a slowly updated prompt teacher. This transforms cross-region dependencies into fine-grained supervision for prompt adaptation. Furthermore, we develop Type-Guided Prompt Learning (TGPL) to impose type-specific residual prediction constraints on different prompt pathways through cyclic mode selection and type-dependent weighting, while preserving joint prompt optimization in classification learning. During training, the ViG backbone remains frozen, and only lightweight adaptation components are optimized; auxiliary components are removed during inference without introducing additional inference-time auxiliary computation. Experiments on 10 visual classification benchmarks show that JEP-VGP improves the average accuracy of VGP by 1.5 percentage points under the same experimental settings.

1 INTRODUCTION

Vision Graph Neural Networks (ViGs) represent local image regions as graph nodes and dynamically construct connections according to node features. By employing graph convolution to model spatially separated yet semantically related regions, ViGs can flexibly capture non-local and irregular semantic relationships, and have demonstrated strong representation capability in tasks such as image classification and object detection (Han et al., 2022).

When adapting pretrained models to different downstream tasks, full fine-tuning requires updating and storing the complete set of model parameters, resulting in substantial adaptation and storage costs. Parameter-efficient fine-tuning optimizes only a small subset of parameters and provides an effective way to reuse pretrained knowledge. Among these approaches, visual prompt learning introduces learnable prompts while keeping the backbone frozen, thereby adjusting feature representations for downstream tasks (Bahng et al., 2022). Visual Prompt Tuning (VPT) demonstrates the effectiveness of this paradigm for Vision Transformers (Jia et al., 2022). For vision graph structures, Vision Graph Prompting (VGP) further employs semantic low-rank decomposition to construct Graph, Edge, and Node Prompts, which influence visual graph representations through virtual-node interaction, neighborhood information propagation, and node feature adjustment, respectively, enabling parameter-efficient adaptation of pretrained ViGs (Ai et al., 2025).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

During such adaptation, prompts influence the final image representation and classification results by modifying node representations and their message passing. Therefore, beyond prompt structure design, how to effectively guide the prompt-induced changes in node representations is also worth investigating. Vision graph prompting methods represented by VGP mainly rely on image-level classification supervision to optimize prompt parameters (Ai et al., 2025). Although the classification loss can update prompts through backpropagation, prompt-induced changes in node representations primarily receive learning feedback through the overall classification outcome and lack direct constraints from visual graph context. Consequently, cross-region dependencies in visual graphs have not yet been explicitly transformed into fine-grained learning signals that act directly on these representation changes. This leads to the central question addressed in this work: *how can cross-region contextual relationships in visual graphs be transformed into fine-grained learning signals for node representation changes to improve downstream prompt adaptation?*

Related studies on the Joint-Embedding Predictive Architecture (JEPA) provide inspiration for this work. I-JEPA predicts target-region representations from visible image context in representation space (Assran et al., 2023), while V-JEPA and V-JEPA 2 extend this predictive paradigm to video representation learning and downstream visual understanding (Bardes et al., 2024; Assran et al., 2025). These studies demonstrate that context-to-target prediction in representation space can transform latent dependencies in visual data into effective learning signals. For vision graph prompting, this suggests a natural way to exploit cross-region dependencies as complementary supervision beyond image-level classification.

Inspired by these observations, a key challenge in introducing contextual prediction into vision graph prompt learning is to determine an appropriate prediction target. If the prompted node representations are predicted directly, the prediction target simultaneously contains representations already provided by the pretrained backbone and changes induced by prompts, making it difficult to characterize the latter separately. Therefore, we focus the prediction target on *prompt-induced changes in node representations*. By explicitly extracting these changes and imposing contextual prediction constraints on them, the resulting learning signals can more directly guide prompt adaptation.

To this end, we propose **JEP-VGP**, a vision graph prompt learning framework inspired by JEPA that takes prompt residuals as prediction targets and provides explicit contextual guidance for prompt-induced changes in node representations. The framework consists of two closely connected modules. Prompt-Residual Embedding Prediction (PREP) explicitly characterizes prompt-induced changes in node representations and constructs a learning task that predicts the residuals of target nodes from visible context. Type-Guided Prompt Learning (TGPL) further applies this prediction constraint to different prompt pathways, providing targeted contextual supervision for different prompt types.

Specifically, under the same prompt configuration, PREP compares visual graph node representations obtained with learnable prompt adjustments enabled and disabled, and defines their difference as the *prompt residual*. This residual describes the node representation changes induced by prompts for the current input, thereby making these changes explicit prediction targets. We then mask the input features of a subset of nodes while preserving their positions in the feature grid and the subsequent graph interactions. A lightweight predictor is employed to predict the prompt residuals of target nodes under the full-feature view from the prompt residuals formed by visible context. The prediction targets are provided by a slowly updated prompt teacher with stop-gradient. Through this prediction task, PREP transforms cross-region dependencies in visual graphs into fine-grained supervision that directly constrains node representation changes, encouraging prompts to exploit contextual information from other regions to learn the representation changes of target nodes under the full-feature view, thereby complementing image-level classification supervision with explicit contextual guidance.

Furthermore, the above prediction constraint needs to be applied to the pathways of different prompt types. Node, Edge, and Graph Prompts influence visual graph representations through node feature adjustment, neighborhood information propagation, and virtual-node interaction, respectively (Ai et al., 2025). When all prompts jointly participate in the auxiliary forward pass, the extracted prompt residual reflects their combined effects, making it difficult for the overall prediction target to correspond explicitly to a specific prompt pathway. To provide targeted learning guidance, TGPL constructs residual prediction tasks separately for different prompt types. At each auxiliary update, one prompt type is selected, and the same mode is maintained across the student, teacher, and their corresponding reference pathways throughout the four stages. During training, Node, Edge, and Graph

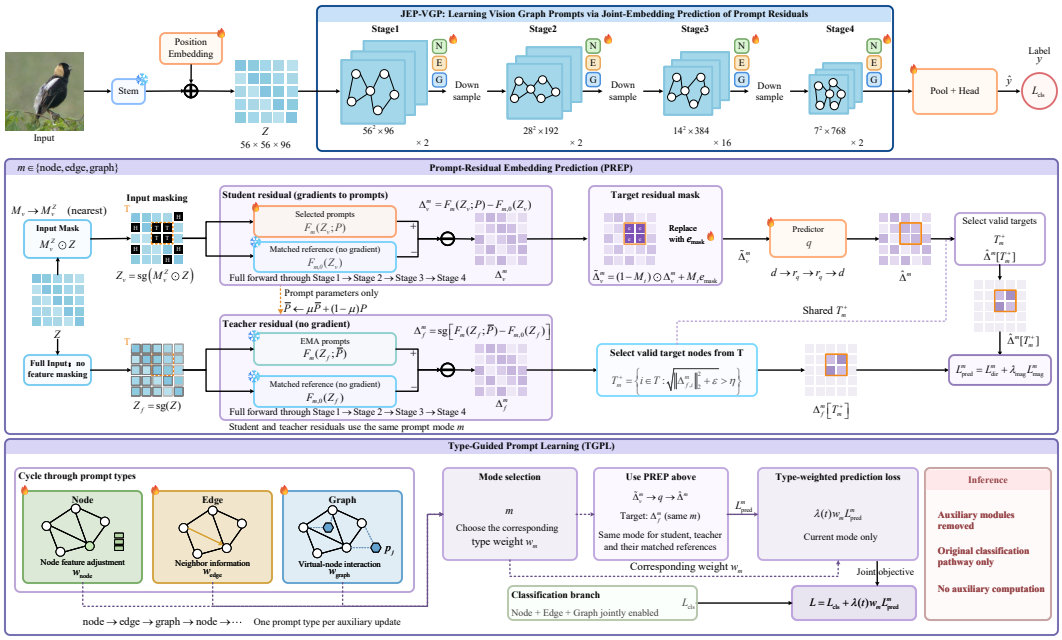


Figure 1: Overview of the JEP-VGP framework. The classification branch jointly activates Node, Edge, and Graph Prompts on the frozen Pyramid ViG backbone. PREP predicts target prompt residuals from masked-view contextual residuals with an EMA-updated teacher, while TGPL organizes type-specific residual prediction tasks with weighted optimization. All residuals are extracted at the end of Stage 4, and auxiliary modules are removed during inference.

Prompts are selected cyclically, and type-dependent weights are used to regulate the contribution of the corresponding prediction tasks. Meanwhile, the classification branch always retains the joint participation of all prompts.

In summary, PREP models prompt-induced node representation changes through residual prediction, while TGPL provides type-specific contextual constraints across prompt pathways. Together, they leverage visual graph context for prompt learning. During training, the ViG backbone remains frozen, and only prompt parameters, positional embeddings, classification head, and lightweight predictor are optimized. The teacher and predictor are removed during inference, leaving the original classification pathway unchanged without additional inference-time auxiliary computation. The main contributions are summarized as follows:

- **We propose JEP-VGP, a vision graph prompt learning framework.** JEP-VGP introduces prompt residual prediction to transform cross-region contextual dependencies in visual graphs into fine-grained supervision for prompt adaptation.
- **We develop PREP and TGPL for context-guided prompt learning.** PREP explicitly models prompt-induced changes in node representations through residual prediction, while TGPL provides type-specific supervision for different prompt pathways.
- **We validate JEP-VGP on 10 visual classification benchmarks.** JEP-VGP consistently outperforms VGP under the same settings, improving average accuracy by 1.5 percentage points without additional inference-time auxiliary computation.

2 METHOD

We propose JEP-VGP, which leverages contextual prediction to constrain prompt-induced node representation changes and provide fine-grained supervision for prompt adaptation under a frozen Pyramid ViG backbone. As shown in Fig. 1, the classification branch jointly activates Node, Edge, and Graph Prompts with image-level supervision. Built upon this branch, Prompt-Residual Embedding Prediction (PREP) extracts prompt-induced residuals and predicts target residuals from masked

contextual residuals, while Type-Guided Prompt Learning (TGPL) applies type-specific contextual constraints across different prompt pathways. We first introduce the formulation of vision graph prompting, followed by PREP and the joint optimization strategy of TGPL.

2.1 ViG and VGP Preliminaries

Pyramid Vision Graph Representation. Given an image $I \in \mathbb{R}^{h\times w\times 3}$, ViG represents local image regions as visual nodes and models their relationships through dynamic graph construction and graph convolution (Han et al., 2022). We adopt Pyramid ViG-M as the frozen backbone, where the initial node features are obtained by

$$Z = \mathrm{Stem}_{W_b}(I) + E_{\mathrm{pos}}, \tag{1}$$

where W_b denotes frozen backbone feature extraction weights and E_{pos} denotes trainable positional embeddings. Features are represented as node matrices and reshaped into two-dimensional grids when convolution, downsampling, or masking operations are performed. For 224×224 inputs, $Z \in \mathbb{R}^{N_0\times d_0}$, where $N_0 = 56 \times 56$ and $d_0 = 96$.

The backbone contains four hierarchical stages with outputs denoted as X_1, X_2, X_3, X_4, whose node dimensions are $X_1 \in \mathbb{R}^{56^2\times 96}, X_2 \in \mathbb{R}^{28^2\times 192}, X_3 \in \mathbb{R}^{14^2\times 384}, X_4 \in \mathbb{R}^{7^2\times 768}$. Prompt residuals are extracted at the end of Stage 4, where the node number and feature dimension are $N = N_4 = 49$ and $d = d_4 = 768$, respectively. The initial feature Z is used to construct auxiliary views, while residual extraction and prediction losses are defined on the final node grid.

Vision Graph Prompting. Following VGP (Ai et al., 2025), we adopt low-rank Graph, Edge, and Node Prompts to adapt the frozen ViG backbone. Node, Edge, and Graph Prompts adjust node features, propagate neighborhood information, and introduce virtual-node interactions, respectively.

Let $x_i \in \mathbb{R}^{1\times d_{\mathrm{loc}}}$ denote the projected node feature, r denote the prompt low-rank dimension, and $P_n, P_e, P_g \in \mathbb{R}^{r\times d_{\mathrm{loc}}}$ denote prompt matrices. The low-rank representation is obtained by

$$a_i = \mathrm{MLP}_s(x_i) \in \mathbb{R}^{1\times r}. \tag{2}$$

The Node Prompt is defined as

$$x_i^{\mathrm{n}} = (1-\alpha)x_i + \alpha a_i P_n. \tag{3}$$

The Edge Prompt aggregates neighboring prompt representations:

$$x_i^{\mathrm{e}} = (1-\beta)x_i^{\mathrm{gc}} + \frac{\beta}{|\,\mathrm{Nbr}(i)|}\sum_{k\in\mathrm{Nbr}(i)} a_k P_e, \tag{4}$$

where x_i^{gc} denotes the graph-convolved feature, $\mathrm{Nbr}(i)$ denotes the dynamic neighborhood, and α, β are fixed coefficients.

The Graph Prompt introduces virtual nodes through

$$p_j = b_j P_g \in \mathbb{R}^{1\times d_{\mathrm{loc}}}, \qquad j = 1, \ldots, J, \tag{5}$$

where $b_j \in \mathbb{R}^{1\times r}$ denotes the learnable low-rank coefficient vector associated with the j-th virtual node, and J denotes the number of virtual nodes. These nodes participate in graph interaction and are removed before classification.

All prompt-related parameters, including prompt matrices, virtual-node representations, and generation parameters, are denoted as P. The classification branch jointly activates all prompts:

$$\widehat{y} = \mathrm{softmax}(\mathrm{Head}(\mathrm{Pool}(X_4)))\,. \tag{6}$$

The classification loss is

$$L_{\mathrm{cls}} = -\sum_{c=1}^{D} y_c \log \widehat{y}_c, \tag{7}$$

where D denotes the number of classes. This image-level objective optimizes prompt parameters, while PREP provides additional contextual constraints on prompt-induced representation changes.

2.2 PROMPT-RESIDUAL EMBEDDING PREDICTION (PREP)

Mode-Consistent Prompt Residuals. Directly predicting prompted node representations entangles the original backbone representations with prompt-induced changes. To explicitly characterize the latter, PREP compares the representations with and without learnable prompt adjustments under the same prompt configuration. Let the prompt mode used by the auxiliary task be

$$m \in \{\text{node}, \text{edge}, \text{graph}\}, \tag{8}$$

and let $F_m(Z; P) \in \mathbb{R}^{N \times d}$ denote the node representations after propagating the initial features Z through all four stages under mode m. Here, P represents all prompt-related parameters, m determines the activated prompt pathway, and the backbone weights remain frozen.

The corresponding reference pathway is denoted as $F_{m,0}(Z)$, where the prompt adjustment is removed while maintaining the same prompt configuration. Specifically, for Node and Edge modes, the fixed mixing coefficients remain unchanged and only prompt adjustment terms are removed; for the Graph mode, the same number of virtual nodes is retained with zero features. The prompt residual is therefore defined as

$$\Delta^m(Z) = F_m(Z; P) - F_{m,0}(Z). \tag{9}$$

This residual explicitly captures the representation changes induced by prompt adjustments after graph interaction and feature transformation. Since visual graphs are dynamically constructed from node features, the residual also incorporates the effects caused by prompt-induced topology variations.

Masked and Full-Feature Views. At the prediction location, we sample a contiguous target region on the two-dimensional grid and denote its node set as T. The area ratio of the target region is sampled from a predefined range $[\rho_{\min}, \rho_{\max}]$, where $\rho = \frac{|T|}{N}$ represents the proportion of target nodes in the final node grid. The remaining nodes are divided into visible context nodes V and hidden nodes H, satisfying

$$\begin{aligned} V \cup T \cup H &= \{1, \ldots, N\}, \\ V \cap T = V \cap H = T \cap H &= \varnothing. \end{aligned} \tag{10}$$

The visible and target masks are defined as

$$(M_v)_i = \mathbf{1}[i \in V], \qquad (M_t)_i = \mathbf{1}[i \in T], \tag{11}$$

where $M_v, M_t \in \{0, 1\}^{N \times 1}$. M_v constructs the student input, whereas M_t identifies the prediction targets.

We reshape M_v to the two-dimensional grid and resize it to the initial feature resolution through nearest-neighbor interpolation, obtaining $M_v^Z \in \{0, 1\}^{N_0 \times 1}$. The student and teacher inputs are respectively

$$Z_v = \text{sg}\big(M_v^Z \odot Z\big), \qquad Z_f = \text{sg}(Z), \tag{12}$$

where sg denotes stop-gradient and $\odot$ denotes element-wise multiplication broadcast along the channel dimension. The masked view preserves node positions while removing the initial features of $T \cup H$, allowing hidden nodes to receive contextual information through graph propagation. The full-feature view provides supervision at the corresponding target locations.

Student Residuals and Teacher Targets. Under the masked view, the student residual is computed as

$$\Delta_v^m = F_m(Z_v; P) - F_{m,0}(Z_v). \tag{13}$$

The prompted pathway participates in gradient computation, while the reference pathway is gradient-free. To generate stable targets, we maintain teacher prompt parameters $\bar{P}$, initialize them from the student parameters, and update them using exponential moving average:

$$\bar{P} \leftarrow \mu \bar{P} + (1 - \mu) P, \tag{14}$$

where μ is the EMA decay coefficient. Only teacher prompt-related parameters are updated by EMA, while other corresponding parameters and buffers are synchronized from the student. The teacher is optimized without gradients and generates target residuals from the full-feature view:

$$\Delta_f^m = \text{sg}\big[F_m(Z_f; \bar{P}) - F_{m,0}(Z_f)\big]. \tag{15}$$

The student, teacher, and reference pathways share the same prompt mode m and extract representations at the same network location, ensuring consistent residual semantics. $\Delta_v^m, \Delta_f^m \in \mathbb{R}^{N\times d}$ represent the student and teacher residuals, respectively. PREP learns to predict the teacher residuals at target locations from contextual student residuals.

Target Residual Masking and Shared Prediction. Although input masking removes target features, target nodes can still receive contextual information through graph propagation. To prevent the predictor from directly accessing target residuals, we apply target masking in the residual space and replace residuals at T with a shared learnable vector $e_{\text{mask}} \in \mathbb{R}^{1\times d}$:

$$\widetilde{\Delta}_v^m = (1 - M_t) \odot \Delta_v^m + M_t e_{\text{mask}}. \tag{16}$$

Here, $(1 - M_t) \odot \Delta_v^m$ preserves residuals outside the target region, while $M_t e_{\text{mask}}$ assigns the same mask vector to all target locations. This operation prevents direct target residual access and encourages prediction from contextual information.

The shared predictor q predicts residuals over the full node grid:

$$\widehat{\Delta}^m = q\Big(\widetilde{\Delta}_v^m\Big) \in \mathbb{R}^{N\times d}. \tag{17}$$

The predictor first normalizes the residual input, compresses channels through a 1×1 convolution, aggregates spatial information using a depthwise convolution, and restores the original feature dimension. Its channel transformation is

$$d \to r_q \to r_q \to d, \tag{18}$$

where r_q is the predictor bottleneck dimension and is independent of the prompt low-rank dimension r. We use single-group GroupNorm without learnable affine parameters, with GELU applied after the dimensionality-reduction and depthwise convolutions. The three prompt modes share the same predictor and mask vector. Although predictions are generated for all nodes, supervision is only applied to selected target locations.

Valid Target Selection and Prediction Loss. A teacher residual with zero magnitude provides no effective prediction direction. Therefore, we select valid targets from T according to the stabilized teacher residual magnitude:

$$T_m^+ = \Big\{ i \in T : \sqrt{\|\Delta_{f,i}^m\|_2^2 + \varepsilon} > \eta \Big\}, \tag{19}$$

where i denotes the node index, $\Delta_{f,i}^m$ denotes the teacher residual, and $\varepsilon > 0$ is a numerical stabilization term. The threshold satisfies $\eta \geq \sqrt{\varepsilon}$ to remove zero-residual targets. In our experiments, η is fixed to $\sqrt{\varepsilon}$ to remove zero-magnitude residuals without introducing an additional tuning parameter. $T_m^+ \subseteq T$ is determined by teacher residuals and is used to select both $\widehat{\Delta}^m[T_m^+]$ and $\Delta_f^m[T_m^+]$. Thus, prediction and supervision are performed at identical locations, with both teacher targets and filtering operations treated as stop-gradient.

Prompt residuals contain both directional and magnitude information. Direction-only constraints cannot preserve residual scale, while magnitude-only matching ignores directional consistency. Therefore, PREP jointly optimizes directional and magnitude constraints. The cosine directional loss is defined as

$$L_{\text{dir}}^m = \frac{1}{|T_m^+|} \sum_{i\in T_m^+} \left(1 - \frac{\langle \widehat{\Delta}_i^m, \Delta_{f,i}^m \rangle}{\sqrt{\|\widehat{\Delta}_i^m\|_2^2 + \varepsilon}\sqrt{\|\Delta_{f,i}^m\|_2^2 + \varepsilon}}\right). \tag{20}$$

To constrain the magnitude of representation changes, we apply logarithmic transformation to stabilized residual magnitudes and optimize them using SmoothL1 loss:

$$L_{\text{mag}}^m = \frac{1}{|T_m^+|} \sum_{i\in T_m^+} \text{SmoothL1}\left(\log\left(1 + \sqrt{\|\widehat{\Delta}_i^m\|_2^2 + \varepsilon}\right), \log\left(1 + \sqrt{\|\Delta_{f,i}^m\|_2^2 + \varepsilon}\right)\right). \tag{21}$$

SmoothL1 uses a unit transition threshold. Both losses are averaged over valid target nodes, and sample indices are omitted for simplicity. The prediction loss for the current mode is

$$L_{\text{pred}}^m = L_{\text{dir}}^m + \lambda_{\text{mag}} L_{\text{mag}}^m, \tag{22}$$

where λ_{mag} balances directional and magnitude constraints. If no valid target node exists in the current batch, L_{pred}^m is set to zero.

2.3 TYPE-GUIDED PROMPT LEARNING AND JOINT OPTIMIZATION (TGPL)

Prompt Type Selection. When all prompts participate in the auxiliary forward pass, the resulting residuals contain the combined effects of different prompt pathways, making type-specific supervision difficult. To provide targeted contextual constraints, TGPL performs auxiliary prediction with one prompt mode activated at each update following the cyclic order

$$\text{node} \rightarrow \text{edge} \rightarrow \text{graph} \rightarrow \text{node} \rightarrow \cdots . \tag{23}$$

For the selected mode m, the student, teacher, and reference pathways maintain the same mode across all four stages. PREP then computes the corresponding residuals Δ_v^m, Δ_f^m, and prediction loss L_{pred}^m. This mode consistency ensures that the auxiliary supervision specifically corresponds to the current prompt adjustment while covering all prompt pathways through cyclic updates. Therefore, TGPL decomposes the auxiliary objective according to prompt pathways, enabling the contextual constraint to match the semantic role of each prompt type.

The prompt selection mechanism is applied only to the auxiliary branch. The classification branch always activates all prompts to preserve joint category-discriminative learning. We introduce fixed type-dependent weights w_m to balance different prediction tasks, where the weights for Node, Edge, and Graph prompts are denoted as $\mathbf{w} = (w_{\text{node}}, w_{\text{edge}}, w_{\text{graph}})$, and w_m is selected according to the current prompt mode $m \in \{\text{node}, \text{edge}, \text{graph}\}$. TGPL shares the predictor with PREP and does not introduce additional type-specific prediction networks. Since prompt generation modules may share parameters across modes, type-specific supervision does not require independent prompt parameters.

Joint Training Objective. The prediction objective of the selected prompt mode is jointly optimized with the classification objective:

$$L = L_{\text{cls}} + \lambda(t) w_m L_{\text{pred}}^m, \tag{24}$$

where w_m denotes the type-dependent weight and $\lambda(t)$ controls the contribution of the auxiliary objective. The coefficient λ_{mag} is used only inside L_{pred}^m to balance directional and magnitude constraints. The maximum auxiliary weight $\lambda_{\max}$ controls the overall contribution of residual prediction, while $\mathbf{w}$ adjusts the relative importance among different prompt types. During each auxiliary update, only the prediction loss of the current mode is computed and optimized together with the classification loss.

Since teacher targets are generated from prompt-induced representation changes, directly applying strong prediction constraints at the beginning may affect prompt adaptation. Therefore, we first optimize prompts through classification supervision and gradually introduce the prediction objective using delayed activation and linear ramp-up:

$$\lambda(t) = \begin{cases} 0, & t < t_s, \\ \lambda_{\max} \min\left(1, \dfrac{t - t_s + 1}{t_r}\right), & t \geq t_s, \end{cases} \tag{25}$$

where t denotes the training epoch indexed from 0, t_s is the activation epoch, $t_r > 0$ is the ramp-up duration, and $\lambda_{\max}$ is the maximum auxiliary loss weight. The activation epoch t_s and ramp-up duration t_r are fixed across all experiments. This strategy enables prompts to first learn task-relevant adaptations and then progressively incorporate contextual residual prediction constraints.

Parameter Updates and Inference Path. During training, the backbone weights W_b remain frozen. The classification loss updates prompt parameters P, positional embeddings E_{pos}, and the classification head, while the prediction loss updates the prompt parameters involved in the current auxiliary pathway, together with the shared predictor and mask vector e_{mask}. Since Z_v and Z_f are stop-gradient inputs, the auxiliary branch does not update positional embeddings. The reference pathways and teacher pathway are gradient-free; teacher prompt parameters are updated through EMA, while other corresponding parameters and buffers are synchronized from the student.

All auxiliary pathways maintain the same prompt mode across four stages, and residuals are extracted only at the end of Stage 4 with a single prediction loss, avoiding stage-wise prediction heads. Although the auxiliary branch introduces additional training computation, it is fully removed during inference, including the teacher, reference pathways, masking operations, predictor, and mode selection. Therefore, JEP-VGP retains the original classification pathway with jointly activated prompts and introduces no additional residual computation or prediction steps during inference.

Table 1: Classification accuracy (%) on ten visual benchmarks using ImageNet-21K pretrained Pyramid ViG-M.

Method	Backbone	DTD	CUB	NABirds	Dogs	Flowers	Food	CIFAR100	CIFAR10	GTSRB	SVHN	Avg.
Full Fine-Tune	ViG-M	73.5	89.0	82.8	81.4	98.5	87.4	89.2	98.6	98.0	91.7	89.0
Linear	ViG-M	66.7	76.2	71.3	71.3	81.4	79.2	67.2	89.4	77.4	65.5	74.6
Adapter	ViG-M	68.3	74.4	76.0	66.8	83.5	84.7	82.6	95.5	93.5	93.1	81.8
VP	ViG-M	67.4	81.1	74.5	72.0	94.3	77.4	78.6	93.1	89.0	83.6	81.1
VPT	ViG-M	71.4	77.3	76.4	73.1	95.3	81.9	76.3	93.2	79.7	82.4	80.7
DAM-VP	ViG-M	71.8	82.6	77.4	74.2	95.9	82.3	81.5	94.9	91.4	85.1	83.7
InsVP	ViG-M	69.8	85.0	78.0	77.0	96.1	84.1	83.3	95.8	87.6	89.4	84.6
VFPT	ViG-M	72.1	82.3	77.2	75.6	95.9	83.2	82.4	95.4	85.4	86.1	83.6
GraphPrompt	ViG-M	63.4	72.5	76.9	69.9	83.6	80.2	81.4	94.2	91.2	92.6	80.6
GPF-Plus	ViG-M	71.0	82.0	77.2	78.2	95.7	82.6	80.9	94.5	90.5	83.1	83.6
VGP	ViG-M	74.8	87.4	80.9	81.7	98.2	89.5	89.7	98.3	98.1	96.9	89.6
JEP-VGP (Ours)	**ViG-M**	**78.9**	**89.6**	**84.7**	**83.6**	**99.4**	**90.6**	**90.3**	**98.5**	**98.4**	**97.1**	**91.1**

Table 2: Ablation study of PREP and TGPL on CUB and Flowers. "Joint" activates all prompt types, "Fixed" uses a single prompt type, and "Cyclic" cyclically selects among prompt types.

Setting	Method	PREP	Type Selection	Node	Edge	Graph	CUB	Flowers
A0	VGP	–	–	–	–	–	87.4	98.2
A1	VGP + PREP	✓	Joint	✓	✓	✓	88.7	98.6
A2	PREP-Node	✓	Fixed	✓	–	–	89.2	99.0
A3	PREP-Edge	✓	Fixed	–	✓	–	89.3	99.2
A4	PREP-Graph	✓	Fixed	–	–	✓	88.9	98.8
A5	**JEP-VGP**	✓	Cyclic	✓	✓	✓	**89.6**	**99.4**

3 EXPERIMENTS

This section systematically evaluates the effectiveness of the proposed JEP-VGP framework for vision graph prompt learning through extensive experiments. First, we compare JEP-VGP with representative parameter-efficient fine-tuning methods on ten visual classification benchmarks to assess its overall classification performance. Furthermore, ablation studies are conducted to analyze the contribution of each component, and sensitivity experiments are performed to investigate the impact of key hyperparameter configurations on model performance.

3.1 DATASETS AND EXPERIMENTAL SETTINGS

Following the experimental protocol of VGP (Ai et al., 2025), we evaluate JEP-VGP on ten representative visual classification datasets, including CIFAR-10, CIFAR-100 (Krizhevsky, 2009), DTD (Cimpoi et al., 2014), CUB200 (Wah et al., 2011), NABirds (Van Horn et al., 2015), Stanford Dogs (Khosla et al., 2011), Oxford Flowers (Nilsback & Zisserman, 2008), Food101 (Bossard et al., 2014), GTSRB (Stallkamp et al., 2012), and SVHN (Netzer et al., 2011), covering texture recognition, fine-grained classification, and general image classification tasks. All experiments adopt the ImageNet-21K pretrained Pyramid ViG-M as the frozen backbone (Han et al., 2022; Deng et al., 2009), with inputs resized to 256×256 and center-cropped to 224×224. During training, only the student prompt parameters, positional embeddings, classification head, and lightweight predictor are optimized, while the remaining backbone parameters are frozen; teacher prompt parameters are updated via EMA without gradient optimization. All models are trained for 100 epochs using AdamW (Loshchilov & Hutter, 2017) with a learning rate of 0.001, weight decay of 0.05, and cosine annealing scheduling. For fair comparison, all methods follow the same backbone, preprocessing strategy, and training configuration.

3.2 COMPARISON METHODS

We compare JEP-VGP with several representative parameter-efficient fine-tuning methods, including VP (Bahng et al., 2022), VPT (Jia et al., 2022), DAM-VP (Huang et al., 2023), InsVP (Liu et al., 2024), VFPT (Zeng et al., 2024), GraphPrompt (Liu et al., 2023), GPF-Plus (Fang et al., 2023), and VGP (Ai et al., 2025). Full Fine-Tune is adopted as a reference baseline. For fair comparison, all methods use Pyramid ViG-M (Han et al., 2022) as the backbone and follow the same training

Table 3: Hyperparameter sensitivity on CUB. Bold denotes the default setting.

Hyperparameter	Candidate Values	Accuracy (%)
$[\rho_{\min}, \rho_{\max}]$	$[0.025, 0.05]/[0.05, 0.10]/[\mathbf{0.10}, \mathbf{0.20}]/[0.15, 0.30]$	$88.9/89.3/\mathbf{89.6}/89.3$
λ_{mag}	$0/0.1/\mathbf{0.25}/0.5/1.0$	$88.5/89.2/\mathbf{89.6}/89.4/89.2$
$\lambda_{\max}$	$0.05/\mathbf{0.5}/1.0/1.5/2$	$89.2/\mathbf{89.6}/89.3/89.3/89.2$
μ	$0.99/0.996/\mathbf{0.999}/0.9995$	$89.4/89.5/\mathbf{89.6}/89.5$
r_q	$8/\mathbf{16}/32/64$	$89.2/\mathbf{89.6}/89.6/89.7$
$\mathbf{w} = (w_{\text{node}}, w_{\text{edge}}, w_{\text{graph}})$	$(1,1,1)/(1,1.5,1)/(1,1,0.75)/(\mathbf{1}, \mathbf{1.5}, \mathbf{0.75})$	$89.3/89.3/89.4/\mathbf{89.6}$

protocol. In particular, the comparison with VGP directly reflects the performance improvement of JEP-VGP in vision graph prompt learning and further evaluates the contribution of the proposed prompt residual prediction and contextual supervision mechanisms to model adaptation.

3.3 Performance Analysis

Table 1 reports the classification results on ten visual benchmarks. JEP-VGP consistently outperforms VGP with the same Pyramid ViG-M backbone, improving the average accuracy from 89.6% to 91.1%, corresponding to a gain of 1.5 percentage points. In particular, it achieves improvements of 4.1 and 3.8 percentage points on DTD and NABirds, respectively. JEP-VGP also surpasses Full Fine-Tune on 9 out of 10 datasets and improves the average accuracy from 89.0% to 91.1%. These results indicate that explicitly imposing contextual prediction constraints on prompt-induced node representation changes enables JEP-VGP to better exploit cross-region dependencies in visual graphs and improve downstream adaptation under a frozen ViG backbone.

3.4 Ablation Study

We evaluate the contributions of PREP and TGPL on CUB and Flowers using the same Pyramid ViG-M backbone and training configuration. As shown in Table 2, adding PREP to VGP (A1) improves accuracy from 87.4%/98.2% to 88.7%/98.6%, indicating that prompt-residual prediction provides effective contextual supervision beyond image-level classification. In A1, all three prompt types are jointly activated in the auxiliary branch to form a combined residual. A2–A4 further apply fixed-type residual prediction to Node, Edge, or Graph Prompts, and all outperform A1, showing that type-specific residual prediction is effective across all three prompt pathways. The full JEP-VGP (A5), which cyclically selects all three prompt types, achieves the best results of 89.6% and 99.4%, improving over A1 by 0.9 and 0.8 percentage points, respectively. These results suggest that decomposing unified residual prediction into type-specific auxiliary tasks better exploits the complementary roles of different prompt pathways.

3.5 Hyperparameter Sensitivity Analysis

Table 3 analyzes the sensitivity of JEP-VGP to key hyperparameters on CUB. The default settings are $[\rho_{\min}, \rho_{\max}] = [0.10, 0.20]$, $\lambda_{\text{mag}} = 0.25$, $\lambda_{\max} = 0.5$, $\mu = 0.999$, $r_q = 16$, and $\mathbf{w} = (1, 1.5, 0.75)$. Overall, JEP-VGP remains stable across different hyperparameter configurations. Moderate target masking and auxiliary-loss weights yield better performance, while the model is relatively insensitive to the EMA decay coefficient μ. Increasing the predictor bottleneck dimension r_q provides only marginal gains, indicating that the default setting offers a favorable trade-off between accuracy and parameter efficiency. The default type-dependent weighting outperforms uniform weighting.

4 Conclusion

We propose JEP-VGP, a vision graph prompt learning framework that uses prompt residual prediction to provide contextual supervision under a frozen backbone. PREP predicts target prompt residuals from masked context, while TGPL imposes type-specific constraints across prompt pathways. Experiments on ten benchmarks show that JEP-VGP improves the average accuracy of VGP by 1.5 percentage points without additional inference-time auxiliary computation.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript translation, software debugging, and language editing for improved clarity and readability. We reviewed all AI-assisted work: AI-assisted code modifications were manually verified and tested, and AI-assisted text was reviewed and revised by the authors. We take responsibility for the final content of this work, including text and code produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the JEP-VGP architecture, training objectives, and optimization procedure in the main paper. The datasets, training settings, and key hyperparameters used in our experiments are reported in Section 3, while additional analysis is provided in the appendix. These details are intended to facilitate reproduction of the reported results.

REFERENCES

Zixiang Ai, Zichen Liu, and Jiahuan Zhou. Vision graph prompting via semantic low-rank decomposition. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 809–821. PMLR, 2025.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems*, 36:52464–52489, 2023.

Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. *Advances in neural information processing systems*, 35:8291–8303, 2022.

Qidong Huang, Xiaoyi Dong, Dongdong Chen, Weiming Zhang, Feifei Wang, Gang Hua, and Nenghai Yu. Diversity-aware meta visual prompting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10878–10887, 2023.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR Workshop on Fine-Grained Visual Categorization*, 2011.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Ning Liao, Bowen Shi, Xiaopeng Zhang, Min Cao, Junchi Yan, and Qi Tian. Rethinking visual prompt learning as masked visual token modeling. *Artificial Intelligence*, 348:104417, 2025. doi: 10.1016/j.artint.2025.104417.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM web conference 2023*, pp. 417–428, 2023.

Zichen Liu, Yuxin Peng, and Jiahuan Zhou. Insvp: Efficient instance visual prompting from image itself. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 6443–6452, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Mustafa Munir, William Avery, Md Mostafijur Rahman, and Radu Marculescu. Greedyvig: Dynamic axial graph construction for efficient vision gnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6118–6127, 2024.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 595–604, 2015.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Runjia Zeng, Cheng Han, Qifan Wang, Chunshu Wu, Tong Geng, Lifu Huangg, Ying Nian Wu, and Dongfang Liu. Visual fourier prompt tuning. *Advances in Neural Information Processing Systems*, 37:5552–5585, 2024.

A Related Work

A.1 Vision Graph Neural Networks

ViGs (Han et al., 2022) represent image regions as graph nodes and model irregularly distributed semantic relationships through dynamic k-nearest neighbor graph construction and graph convolution. Subsequent studies further improve the architecture and graph construction efficiency of visual graphs. For example, GreedyViG (Munir et al., 2024) reduces the computational cost of neighbor search through dynamic axial graph construction. These studies mainly focus on improving the representation capability and computational efficiency of visual graph backbones. In contrast, this work investigates prompt learning under frozen pretrained ViGs and exploits visual graph context to guide prompt-induced changes in node representations.

A.2 Visual Prompt Learning

Visual prompt learning enables downstream adaptation of frozen pretrained models by optimizing only a small number of learnable parameters (Bahng et al., 2022). VPT (Jia et al., 2022) introduces learnable prompts into Vision Transformers, while VPTM (Liao et al., 2025) reformulates downstream classification as masked visual token modeling to better align pretraining and adaptation objectives. For visual graph structures, VGP (Ai et al., 2025) employs low-rank Graph, Edge, and Node Prompts to achieve representation adjustment through different pathways. However, it mainly relies on image-level classification supervision and lacks direct contextual constraints on prompt-induced changes in node representations. JEP-VGP takes prompt residuals as prediction targets, exploits cross-region dependencies to provide fine-grained supervision, and imposes prediction constraints on different prompt pathways according to prompt types.

Table 4: Parameter and computational complexity comparison across ten visual datasets. All methods use an ImageNet-21K-pretrained Pyramid ViG-M backbone with 224×224 inputs. Param. denotes trainable parameters, and Infer. denotes the forward computational cost for classification inference.

Methods	Backbone	Metric	*DTD*	*CUB*	*NABirds*	*Dogs*	*Flowers*	*Food*	*CIFAR-100*	*CIFAR-10*	*GTSRB*	*SVHN*	**Average**
Full Fine-Tune	ViG-M	Param. (M)	48.55	48.71	49.54	48.62	48.61	48.60	48.60	48.51	48.54	48.51	$48.68_{(100.00\%)}$
		Infer. (G)	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	$8.94_{(100.00\%)}$
Linear	ViG-M	Param. (M)	1.14	1.30	2.13	1.21	1.20	1.19	1.19	1.10	1.13	1.10	$1.27_{(-97.40\%)}$
		Infer. (G)	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	$8.94_{(+0.00\%)}$
Adapter	ViG-M	Param. (M)	1.28	1.44	2.27	1.35	1.34	1.33	1.33	1.24	1.28	1.24	$1.41_{(-97.10\%)}$
		Infer. (G)	8.98	8.98	8.98	8.98	8.98	8.98	8.98	8.98	8.98	8.98	$8.98_{(+0.40\%)}$
VP	ViG-M	Param. (M)	1.21	1.37	2.20	1.28	1.26	1.26	1.26	1.17	1.20	1.17	$1.34_{(-97.20\%)}$
		Infer. (G)	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	$8.94_{(+0.00\%)}$
VPT	ViG-M	Param. (M)	1.25	1.41	2.24	1.33	1.31	1.31	1.31	1.22	1.25	1.22	$1.38_{(-97.20\%)}$
		Infer. (G)	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	$9.09_{(+1.70\%)}$
DAM-VP	ViG-M	Param. (M)	2.88	3.04	3.87	2.96	2.94	2.94	2.94	2.85	2.88	2.85	$3.02_{(-93.80\%)}$
		Infer. (G)	17.88	17.88	17.88	17.88	17.88	17.88	17.88	17.88	17.88	17.88	$17.88_{(+100.00\%)}$
InsVP	ViG-M	Param. (M)	1.32	1.48	2.31	1.40	1.38	1.38	1.38	1.28	1.32	1.28	$1.45_{(-97.00\%)}$
		Infer. (G)	13.18	13.18	13.18	13.18	13.18	13.18	13.18	13.18	13.18	13.18	$13.18_{(+47.40\%)}$
VFPT	ViG-M	Param. (M)	1.25	1.41	2.24	1.33	1.31	1.31	1.31	1.22	1.25	1.22	$1.38_{(-97.20\%)}$
		Infer. (G)	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	$9.09_{(+1.70\%)}$
GraphPrompt	ViG-M	Param. (M)	1.14	1.30	2.13	1.21	1.20	1.19	1.19	1.10	1.14	1.10	$1.27_{(-97.39\%)}$
		Infer. (G)	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	$8.94_{(+0.00\%)}$
GPF-Plus	ViG-M	Param. (M)	1.14	1.30	2.13	1.21	1.20	1.20	1.19	1.10	1.14	1.10	$1.27_{(-97.39\%)}$
		Infer. (G)	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	8.94	$8.94_{(+0.04\%)}$
VGP (Backbone)	ViG-M	Param. (M)	2.47	2.63	3.46	2.55	2.53	2.53	2.53	2.44	2.47	2.44	$2.61_{(-94.60\%)}$
		Infer. (G)	9.22	9.22	9.22	9.22	9.22	9.22	9.22	9.22	9.22	9.22	$9.22_{(+3.10\%)}$
JEP-VGP (Ours)	ViG-M	Param. (M)	2.50	2.66	3.49	2.57	2.56	2.56	2.55	2.46	2.50	2.46	$2.63_{(-94.60\%)}$
		Infer. (G)	9.22	9.22	9.22	9.22	9.22	9.22	9.22	9.22	9.22	9.22	$9.22_{(+3.19\%)}$

A.3 Joint-Embedding Predictive Architecture

Joint-Embedding Predictive Architectures learn dependencies within visual data through context-to-target prediction in representation space. I-JEPA (Assran et al., 2023) predicts target region repre-

sentations from visible image context; V-JEPA (Bardes et al., 2024) extends feature prediction to video learning; and V-JEPA 2 (Assran et al., 2025) further explores this paradigm for video understanding, action anticipation, and robot planning. Inspired by these studies, JEP-VGP focuses the prediction target on prompt-induced changes in node representations. It predicts the residuals of target nodes under the full-feature view from prompt residuals formed by visible context, providing explicit contextual guidance for prompt adaptation under a frozen backbone.

B Complexity Analysis

Table 4 compares the trainable parameters and inference computational cost of JEP-VGP with Full Fine-Tune and representative parameter-efficient adaptation methods across ten visual datasets. Full Fine-Tune requires an average of 48.68M trainable parameters, whereas JEP-VGP requires only 2.63M, corresponding to a 94.60% reduction in trainable parameters. Compared with the VGP backbone, which requires 2.61M trainable parameters on average, JEP-VGP introduces only 0.02M additional parameters, indicating that the proposed framework incurs only a marginal increase in the trainable parameter budget. In terms of inference computation, JEP-VGP requires 9.22G FLOPs, the same as VGP, showing that the proposed training-time auxiliary components do not introduce additional computational overhead during inference. Moreover, its inference cost remains substantially lower than that of DAM-VP (17.88G) and InsVP (13.18G). Overall, JEP-VGP retains the parameter-efficient and inference-efficient characteristics of VGP while introducing only a negligible increase in trainable parameters.