I2LP: Iterative ILP with Update-Size Control for Reducing Surrogate–Task Mismatch in Bit-Width Selection

Shinya Gongyo[*] **Ryosuke Ogasawara**[*] **Tatsuya Moe**[*]

Masafumi Mori[†] **Yusuke Sekikawa**[*] **Mitsuru Ambai**[*]

gongyo.shinya@core.d-itlab.co.jp

Abstract

Mixed-precision quantization improves the accuracy–efficiency trade-off by assigning different bit-widths across a network. Since bit-width selection is a combinatorial problem, existing methods often optimize additive surrogate losses instead of directly evaluating task losses in all configurations. We show that these surrogates can suffer from surrogate–task mismatch, especially when many layers are changed simultaneously from a reference configuration. This mismatch can arise even when the surrogate is constructed from directly evaluated single-layer task-loss changes. We further observe that, although allowing more layers to change improves the surrogate objective, the evaluated task loss can be minimized at an intermediate limit on the number of changed layers. Based on this observation, we propose Iterative Integer Linear Programming (I2LP), which iteratively refines a reference bit configuration. At each refinement step, I2LP solves multiple ILPs with different limits on how many layers may change from the current reference, and accepts the best candidate only when it reduces the task loss. I2LP applies to both post-training quantization and quantization-aware training across a variety of vision models, consistently outperforming uniform-precision baselines and existing mixed-precision methods. We further demonstrate its applicability to weight-only quantization of a large language model, improving perplexity and zero-shot task accuracy over uniform-precision quantization. Code is available at https://github.com/DensoITLab/I2LP.

1 Introduction

Deep neural networks achieve strong performance on vision tasks such as image recognition [8, 12, 31] and object detection [27, 28]. However, their memory footprint and computational cost make deployment on resource-constrained devices challenging. Quantization reduces this cost by replacing floating-point weights and activations with low-precision representations.

While uniform-precision quantization has been widely studied [15, 34, 25, 36], quantization sensitivity varies substantially across layers. This suggests that selecting bit-widths at the layer or substructure level can improve the accuracy–efficiency trade-off.

However, the resulting bit-width selection, hereafter bit selection, is a combinatorial optimization problem. Even a binary choice between 4-bit and 8-bit precision across 18 selectable layers in

[*]DENSO IT Laboratory, Tokyo, Japan
[†]DENSO CORPORATION, Aichi, Japan

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

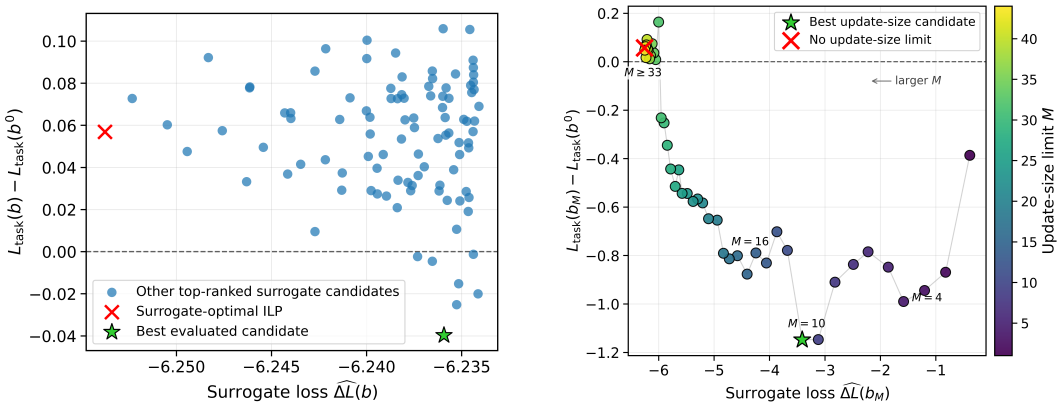


Figure 1: **Motivation for introducing update-size control.** Results on MobileNet-V2 under the uniform-4-bit-equivalent BOPs budget, using the uniform 4-bit configuration as the reference b^0. **(a)** Among top-ranked surrogate candidates, the surrogate-optimal ILP solution increases the evaluated task loss relative to b^0, while another candidate substantially reduces it. **(b)** As the update-size limit M increases, the surrogate loss is monotonically non-increasing, whereas the evaluated task loss is non-monotonic in this experiment and is minimized at an intermediate M. The red crosses in (a) and (b) denote the same surrogate-optimal ILP solution, which corresponds to solving the ILP without an update-size limit.

ResNet-18 yields 2^{18} possible configurations, and each candidate requires evaluating the task loss. Exhaustive search is therefore computationally infeasible.

To address this challenge, many mixed-precision methods replace the original combinatorial problem with an additive surrogate loss. In this formulation, each candidate bit-width is assigned a layer-wise score, such as a Hessian-based sensitivity [7] or an orthogonality-based sensitivity [22], and a single integer linear program (ILP) is solved under a resource constraint [14, 41, 3, 22, 1, 17]. This strategy is computationally attractive because the resulting objective and constraints are linear. However, it relies on a strong additivity assumption: the effect of changing multiple layers is approximated by the sum of independently estimated layer-wise effects. Accurate single-layer effects therefore do not imply an accurate additive model of joint changes.

To make this distinction precise, consider an additive surrogate based on single-layer task-loss changes around a reference bit configuration $\mathbf{b}^0 = [b_1^0, \ldots, b_N^0]$ over N selectable substructures [14]. Let L_{task} denote the task loss evaluated on calibration data, including distillation losses such as KL divergence, and let $\mathbf{b}_i^0$ denote the configuration obtained from $\mathbf{b}^0$ by replacing only its i-th entry b_i^0 with b_i: $\mathbf{b}_i^0 = [b_1^0, \ldots, b_{i-1}^0, b_i, b_{i+1}^0, \ldots, b_N^0]$. We define

$$\Delta L_i(b_i) = L_{\text{task}}(\mathbf{b}_i^0) - L_{\text{task}}(\mathbf{b}^0),$$

and construct the additive surrogate

$$\widehat{\Delta L}(\mathbf{b}) = \sum_{i=1}^{N} \Delta L_i(b_i).$$

Minimizing this surrogate under a resource budget yields the surrogate-optimal ILP solution. However, minimizing the surrogate loss is not equivalent to minimizing the evaluated task loss.

Figure 1(a) illustrates this failure on MobileNet-V2 under the same 4-bit BOPs budget. Even among configurations ranked highly by the surrogate loss, the surrogate-optimal ILP solution (red cross mark) increases the evaluated task loss relative to the reference $\mathbf{b}^0$, whereas another top-ranked candidate substantially reduces it (green star mark). Thus, the issue is not merely that the surrogate optimum is suboptimal under the evaluated task loss, but directly that minimizing the additive surrogate can even select a configuration that degrades the reference.

To make the origin of this mismatch clear, we express the task-loss change using an exact discrete interaction decomposition around the reference configuration:

$$L_{\text{task}}(\mathbf{b}) - L_{\text{task}}(\mathbf{b}^0) = \widehat{\Delta L}(\mathbf{b}) + \sum_{\substack{i,j \in S(\mathbf{b}) \\ i<j}} I_{ij}(b_i, b_j) + \sum_{\substack{i,j,k \in S(\mathbf{b}) \\ i<j<k}} I_{ijk}(b_i, b_j, b_k) + \cdots,$$

where $S(\mathbf{b}) := \{i \mid b_i \neq b_i^0\}$ is the set of layers whose bit-widths differ from the reference configuration $\mathbf{b}^0$, and $I_{ij}, I_{ijk}, \ldots$ denote pairwise and higher-order interaction terms caused by changing multiple layers simultaneously. These interaction terms are defined recursively by subtracting all lower-order effects from the corresponding joint task-loss changes. The decomposition is exact and requires no smoothness assumptions. The additive surrogate retains only the first-order terms, and thus the surrogate–task mismatch is precisely the sum of the omitted interaction terms.

This expansion also clarifies the role of the update-size limit. If at most M layers are changed simultaneously, all interaction terms of order greater than M are absent. In particular, the additive surrogate is exact for one-layer updates, while larger update sizes introduce progressively higher-order interactions that are not captured by the surrogate. Thus, limiting the update size can improve the alignment between surrogate loss and evaluated task loss by suppressing the accumulation of ignored interaction terms.

Figure 1(b) visualizes this trade-off. As M increases, the feasible set expands, so the optimized surrogate loss can only decrease or remain unchanged. At the same time, larger updates can activate more interaction terms, making the surrogate less reliable as a predictor of the evaluated task loss. Consequently, the evaluated task loss is non-monotonic in M, and its minimum is attained at an intermediate update-size limit (green star mark). Notably, the best candidate in Fig. 1(b) is substantially better than the best evaluated top-ranked surrogate candidate in Fig. 1(a), showing that update-size control is more effective than simply evaluating many high-ranked no-limit surrogate candidates.

Based on this observation, we propose Iterative Integer Linear Programming (I2LP), a bit-selection framework that repeatedly solves update-size-constrained ILPs around the current reference configuration (Figure 2). At each iteration, I2LP computes single-layer task-loss changes around the current reference and solves a set of update-size-constrained ILPs, one for each update-size limit $M \in \mathcal{U}$. In the ILP for M, the solution is constrained to differ from the current reference in at most M layers.

The resulting candidates are evaluated by the task loss, and the reference is updated only if the best candidate reduces this loss. This procedure avoids relying on a single global surrogate optimum and instead refines the reference through a sequence of evaluated, update-size-controlled steps.

We further introduce reconstruction-aware I2LP (recI2LP), which computes the surrogate after reconstruction to align bit selection with the post-reconstruction objective.

Our main contributions are summarized as follows.

- **Surrogate–task mismatch analysis for update-size control.** We analyze the mismatch between additive surrogate loss and task loss through multi-layer interaction terms, showing that even directly evaluated single-layer loss changes do not necessarily yield accurate predictions for joint changes. This motivates controlling the number of changed layers as a way to mitigate the mismatch.
- **Iterative ILP for bit selection.** We propose I2LP, an iterative mixed-precision bit-selection method that solves a set of update-size-constrained ILPs around a reference configuration using single-layer task-loss differences and selects candidates based on evaluated task loss. We further introduce recI2LP, a reconstruction-aware variant for PTQ.
- **Evaluation across quantization settings.** Across PTQ and QAT on CNNs, ViTs, and DETR, I2LP improves the accuracy–efficiency trade-off under BOPs and model-size constraints. We further demonstrate its applicability to weight-only quantization of selected linear layers in a large language model.

2 Related Work

Quantization. Neural network quantization is commonly studied in quantization-aware training (QAT) [2, 4, 9, 15, 21, 25] and post-training quantization (PTQ) [24, 18, 34]. QAT retrains quantized networks and can achieve strong low-bit accuracy, but requires a full labeled dataset and substantial training time. In contrast, PTQ calibrates a pretrained model with a small unlabeled dataset, making it more practical for deployment but more sensitive to aggressive low-bit quantization. Recent PTQ methods improve either quantization parameter calibration, including ViT-oriented methods such as RepQViT and AdaLog [19, 36], or reconstruction-based optimization such as AdaRound, BRECQ, and QDrop [24, 18, 34] in CNN and FIMA-Q and APHQ-ViT in ViT [37, 38].

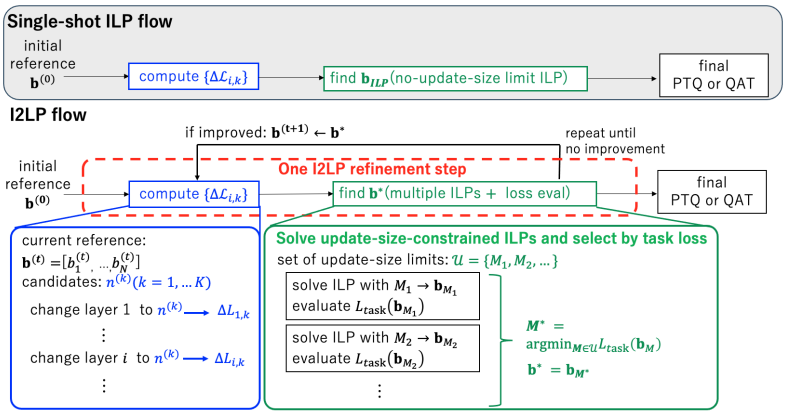


Figure 2: Overview of single-shot ILP and I2LP. The single-shot flow computes surrogate terms once and solves one no-update-size-limit ILP. I2LP instead iteratively refines the reference: at each step, it computes surrogate terms, solves update-size-constrained ILPs, selects by task loss, and updates only if the task loss improves.

Optimization-based Mixed Precision. Beyond metric-based bit selection mentioned above, mixed-precision quantization has been studied with reinforcement learning [35, 33] and differentiable architecture parameters [32, 40, 16]. These methods can model task-loss effects more directly but often require large datasets and heavy computation because the search space grows exponentially with the number of layers.

3 Method

3.1 Problem setup

Consider a network divided into N selectable substructures, such as layers. For each substructure i, we choose one candidate bit assignment from a finite set $\mathcal{B} = \{n^{(1)}, \dots, n^{(K)}\}$ (e.g., $\mathcal{B} = \{2, 3, \dots, 8\}$). Let $\mathbf{b} = [b_1, \dots, b_N] \in \mathcal{B}^N$ denote a bit configuration.

We denote by $C(\mathbf{b})$ the resource cost of configuration $\mathbf{b}$, such as model size or BOPs. If candidate $n^{(k)}$ is assigned to substructure i, we denote its resource cost by $c_{i,k}$. For memory, $c_{i,k}$ is given by the number of parameters at the i-th substructure times the weight bit-widths, and for BOPs, $c_{i,k}$ is computed from the weight and activation bit-widths, and the MAC count of the i-th substructure.

The bit-selection problem is formalized as

$$\min_{\mathbf{b} \in \mathcal{B}^N} \; L_{\text{task}}(\mathbf{b}) \quad \text{s.t.} \quad C(\mathbf{b}) \le C_{\max}. \tag{1}$$

Directly solving this problem is infeasible because evaluating $L_{\text{task}}(\mathbf{b})$ for all configurations is combinatorial.

3.2 Update-size-constrained surrogate ILP

As discussed in Section 1, directly optimizing an additive surrogate over a large update can amplify the surrogate–task mismatch. We therefore restrict the number of substructures whose bit assignments may differ from a reference configuration. Following the notation in Section 1, let $\mathbf{b}^0 = [b_1^0, \dots, b_N^0]$ be the reference configuration and $\mathbf{b}$ be a candidate configuration. We define $S(\mathbf{b}) = \{i \mid b_i \neq b_i^0\}$. For an update-size limit M, we consider

$$\min_{\mathbf{b} \in \mathcal{B}^N} \quad \sum_{i=1}^{N} \Delta L_i(b_i) \quad \text{s.t.} \quad C(\mathbf{b}) \le C_{\max}, \; |S(\mathbf{b})| \le M \tag{2}$$

where $\Delta L_i(b_i) \equiv L_{\text{task}}(\mathbf{b}_i^0) - L_{\text{task}}(\mathbf{b}^0)$ with $\mathbf{b}_i^0 \equiv [b_1^0, \dots, b_{i-1}^0, b_i, b_{i+1}^0, \dots, b_N^0]$.

We introduce binary variables $x_{i,k} \in \{0, 1\}$, where $x_{i,k} = 1$ indicates that candidate $n^{(k)}$ is assigned to substructure i. For a given update-size limit M, the update-size-constrained surrogate ILP is

Algorithm 1 I2LP: Iterative ILP for Bit Selection

Input: candidate set $\mathcal{B}$, refinement update-size set $\mathcal{U}_{\text{multi}}$, budget $C_{\max}$, initial high-bit reference $\mathbf{b}^{(0)}$
Output: bit configuration $\mathbf{b}^*$
$t \leftarrow 0, \quad \ell_{\text{ref}} \leftarrow +\infty$
repeat
  $\mathcal{U}^{(t)} \leftarrow \{N\}$ if $t = 0$, otherwise $\mathcal{U}_{\text{multi}}$
  Compute surrogate terms $\{\Delta\mathcal{L}_{i,k}(\mathbf{b}^{(t)})\}$
  for each $M \in \mathcal{U}^{(t)}$ **do**
    $\mathbf{b}_M^{(t)} \leftarrow$ solve Equation (3) with reference $\mathbf{b}^{(t)}$ and update-size limit M
    Evaluate $L_{\text{task}}(\mathbf{b}_M^{(t)})$
  end for
  $M^\star \leftarrow \arg\min_{M \in \mathcal{U}^{(t)}} L_{\text{task}}(\mathbf{b}_M^{(t)})$
  $\ell^\star \leftarrow L_{\text{task}}(\mathbf{b}_{M^\star}^{(t)})$
  if $\ell^\star < \ell_{\text{ref}}$ **then**
    $\mathbf{b}^{(t+1)} \leftarrow \mathbf{b}_{M^\star}^{(t)}$
    $\ell_{\text{ref}} \leftarrow \ell^\star$
    $t \leftarrow t + 1$
  else
    break
  end if
until convergence
return $\mathbf{b}^* \leftarrow \mathbf{b}^{(t)}$

rewritten as

$$\begin{aligned} \min_{\{x_{i,k}\}} \quad & \sum_{i=1}^{N}\sum_{k=1}^{K} \Delta\mathcal{L}_{i,k} x_{i,k} \\ \text{s.t.} \quad & \sum_{k=1}^{K} x_{i,k} = 1,\ \forall i, \qquad \sum_{i=1}^{N}\sum_{k=1}^{K} c_{i,k} x_{i,k} \le C_{\max}, \\ & \sum_{i=1}^{N} x_{i,k_0^i} \ge N - M, \qquad x_{i,k} \in \{0,1\},\ \forall i,k. \end{aligned} \tag{3}$$

where $\Delta\mathcal{L}_{i,k} \equiv \Delta L_i(b_i = n^{(k)})$ and k_0^i specifies the index for the reference bit-width configuration. The constraints enforce one bit-width per substructure, the resource budget, and the update-size limit, which keeps at least $N - M$ substructures unchanged from $\mathbf{b}^0$. When $M = N$, this constraint is inactive, and the ILP reduces to the usual surrogate ILP without an update-size limit. Since both the objective and the constraints are linear in the binary decision variables $\{x_{i,k}\}$, this problem can be solved using standard ILP solvers.

For $M_1 \le M_2$, the feasible set of the ILP with update-size limit M_1 is contained in that with M_2. Therefore, the optimal surrogate loss is monotonically non-increasing in M.

3.3 I2LP algorithm

The overview of I2LP is illustrated in Figure 2, and its algorithm is outlined in Algorithm 1. At iteration t, I2LP computes surrogate terms around $\mathbf{b}^{(t)}$, solves ILPs for $M \in \mathcal{U}^{(t)}$, evaluates the candidates by task loss, and accepts the best one only if it improves ℓ_{ref}. The update-size limit M can be viewed as a Hamming-radius constraint, but I2LP selects its effective value by task-loss evaluation over multiple candidates rather than fixing it a priori (see Appendix F). Since the initial high-bit reference $\mathbf{b}^{(0)}$ may violate the target budget, we set $\ell_{\text{ref}} = +\infty$ and use $\mathcal{U}^{(0)} = \{N\}$ to obtain the first budget-feasible reference. Subsequent accepted refinement steps strictly decrease the evaluated calibration task loss.

Update of step size and zero point. The task loss after a bit-width change depends not only on the selected bit-widths but also on the quantization step sizes and zero points. Before evaluating $\Delta L_{i,k}$, we use MSE-calibrated quantization parameters for the corresponding bit-width, either reusing precomputed values (static mode) or recalibrating the modified substructure (dynamic-rough and dynamic-fine modes). Details are provided in Appendix E.

Integration with reconstruction. For reconstruction-based PTQ, the relevant objective is the task loss after reconstruction. Therefore, recI2LP computes $\Delta\mathcal{L}_{i,k}$ after applying reconstruction from the modified block onward. This better aligns the surrogate with the post-reconstruction objective; we provide diagnostic evidence in Appendix D.

Computational cost. For vanilla I2LP, each refinement step requires NK task-loss evaluations to construct the surrogate terms $\{\Delta\mathcal{L}_{i,k}\}$, plus $|\mathcal{U}|$ additional evaluations to select among the ILP candidates. For recI2LP, constructing the surrogate additionally includes reconstruction. In practice, the wall-clock time is dominated by surrogate construction and candidate evaluation, while ILP solving is negligible in our settings. Measured runtimes and solver statistics are reported in Sec. 4.3 and Appendix H.

4 Experiments

We implement our method in PyTorch and conduct experiments on the ImageNet dataset [5] for image classification and COCO 2017 for object detection. We evaluate the effect of mixed precision in both PTQ and QAT settings.

Implementation details. We perform per-layer bit-width selection using a small unlabeled calibration set: 1024 samples for ImageNet classification and 64 samples for COCO object detection. The task loss is evaluated as a distillation loss with temperature 1 between the full-precision and quantized networks. We use an initial high-bit reference, uniform 8-bit for classification and uniform 16-bit for detection. The first pass uses $\mathcal{U}^{(0)} = \{N\}$ to obtain a budget-feasible reference, and subsequent passes use $\mathcal{U}_{\text{multi}} = \{2, 4, \ldots, 2^{\lfloor \log_2 N \rfloor}, N\}$. Candidate bit-widths are $\mathcal{B} = \{2, 3, 4, 5, 6, 7, 8\}$ for classification and $\mathcal{B} = \{2, 4, 8, 16\}$ for detection. Additional details are provided in Appendix C.

4.1 Evaluation in PTQ

We evaluate PTQ on image classification (ResNet-18, MobileNet-V2, and DeiT-T) and object detection (DETR-R50) to demonstrate the generality of our bit-selection method. We compare uniform-precision and mixed-precision PTQ methods with our method both without reconstruction (I2LP) and with reconstruction (recI2LP). Table 1 demonstrates that our method consistently improves the accuracy–efficiency trade-off across model-size and BOPs resource constraints.

4.1.1 Image Classification

CNN. To isolate the effect of bit selection, we compare vanilla I2LP with its uniform-bit counterpart using the same MSE-based step-size/zero-point initialization, denoted MSEinit. As shown in Table 1, changing only the bit configuration with I2LP substantially improves the 4-bit ResNet-18 and MobileNet-V2 results at slightly lower BOPs. For recI2LP, we integrate QDrop [34], a well-established reconstruction method for CNNs. As shown in Table 1, recI2LP consistently surpasses the SOTA uniform-precision methods. Specifically, recI2LP achieves 1.1–1.8 pp improvements over 4/4-bit QDrop on ResNet-18 and MobileNet-V2 at slightly lower model sizes and BOPs. Furthermore, our method consistently outperforms the SOTA mixed-precision methods at slightly lower model sizes and BOPs. In particular, recI2LP surpasses EMQ by up to 1.24 pp in the model-size comparisons.

ViT. For DeiT, we build RepQViT+ as a stronger uniform-precision baseline by retaining the scale reparameterization of RepQViT [19] and replacing its $\log\sqrt{2}$ post-softmax quantizer with nuLSQ [10]. This better captures the non-uniform distribution of post-softmax activations. Table 1(c) shows that, across DeiT-T/S/B and BOPs corresponding to the 3-, 4-, and 6-bit regimes, I2LP consistently achieves higher accuracy than RepQViT+ at lower BOPs.

Comparison with LAMPQ. LAMPQ [17] uses a type-aware Fisher-based metric for ILP-based bit allocation, followed by pairwise one-bit refinements. For a direct comparison, we evaluate I2LP with AdaLog and LAMPQ results under the corresponding AdaLog-based quantization setup and parameter-weighted average-bit budgets. As shown in Table 2, I2LP outperforms both baselines in all six evaluated settings.

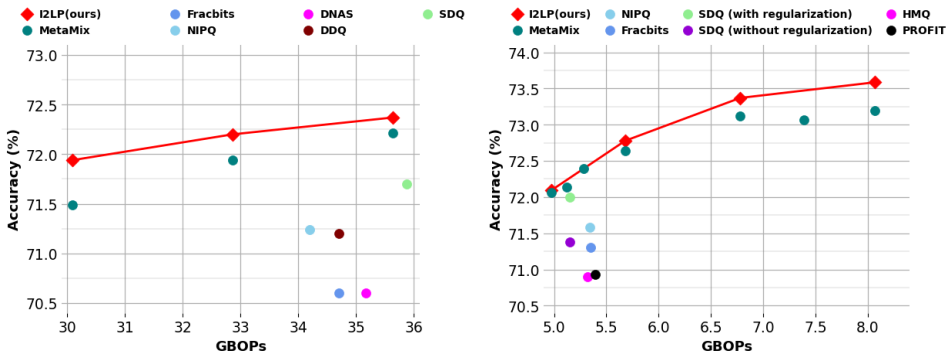


Figure 3: Comparison of mixed-precision QAT methods on ResNet-18 (left) and MobileNet-V2 (right). Compared methods: FracBits-PACT [40], HAWQ-v3 [41], PROFIT [26], DNAS [35], DDQ [42], NIPQ [29], SDQ [13], and MetaMix [16].

4.1.2 Object detection

DETR. Table 3 reports bit-selection results for DETR-R50 on COCO 2017 under BOPs corresponding to the 5-, 6-, and 8-bit regimes. DETR is sensitive to quantization, showing a noticeable drop even at 8-bit. I2LP mitigates this degradation and nearly recovers original AP at slightly lower BOPs. Under more aggressive quantization (6/6 and 5/5), the uniform MSE-initialized counterpart collapses, whereas I2LP changes only the bit configuration and maintains strong detection accuracy, improving AP by 21.6 and 32.2 pp at the corresponding BOPs, respectively. Similar gains are observed across AP_{50}, AP_{75}, and object sizes (AP_s/AP_m/AP_l), demonstrating the robustness of I2LP across multiple BOPs.

4.1.3 Scalability to DINOv3 backbones

We further evaluate I2LP on modern DINOv3 backbones [30], using the standard ImageNet-1K linear-probe protocol. We keep the linear classifier in full precision and apply the quantization and bit-selection only to the frozen pretrained backbone. As shown in Table 4, I2LP consistently improves over uniform 6-bit quantization under matched BOPs budgets, demonstrating its effectiveness across modern vision backbones of different scales.

4.1.4 Weight-only Quantization of a Large Language Model

We further evaluate I2LP on 216 selected linear layers in Qwen3-8B using weight-only quantization. Under a parameter-weighted 3-bit budget over these selected weights, I2LP reduces WikiText-2 perplexity from 41.52 to 18.12 and improves six-task average accuracy from 51.77% to 57.64% compared with uniform 3-bit quantization. Other modules and activations remain in BF16. Additional results and experimental details are provided in Appendix K.

4.2 Evaluation in QAT

We evaluate mixed-precision QAT on ResNet-18 and MobileNet-V2 under a wide range of BOP constraints. After layer-wise bit selection on unlabeled calibration data, we perform QAT with the full labeled dataset using LSQ [9] on the selected bit configuration. Following the strategy of MetaMix [16], the current SOTA mixed-precision approach in QAT, we use ResNet-101 as the teacher network for ResNet-18, and MobileNet-V2 with a $1.2\times$ width multiplier as the teacher network for the original MobileNet-V2. As shown in Figure 3, I2LP consistently outperforms existing methods. Additional comparisons with recent methods, including InfoQ [1] and MSQ [11] at their reported resource settings, are provided in Appendix J.

4.3 Ablation studies

Behavior of iterative updates. Figure 4 summarizes iterative refinement. I2LP achieves lower task loss than the no-update-size-limit ILP across BOPs budgets and converges in a small number

Table 1: Comparison of uniform- and mixed-precision PTQ methods on ResNet-18, MobileNet-V2, and DeiT. † uses 8-bit for the first/last layers, * denotes our implementation. "REC": reconstruction, "MP": mixed precision, "MS": model size, "BOPs": bit operations, "Top-1": accuracy (%).

(a) Results on ResNet-18

Accuracy vs. model size

Method	REC	W/A	MS (MiB)	Top-1
FP32	-	32/32	44.6	72.54
QDrop† [34]	✓	4/4	5.81	69.62
OMPQ† [22]	✓	MP/4	5.5	69.38
EMQ† [6]	✓	MP/4	5.5	70.12
recI2LP (ours)*†	✓	MP/4	5.45	**70.84**
BRECQ† [18]	✓	MP/8	4.0	68.82
OMPQ† [22]	✓	MP/8	4.0	69.41
EMQ† [6]	✓	MP/8	4.0	69.92
recI2LP (ours)*†	✓	MP/8	3.999	**70.98**

Accuracy vs. BOPs

Method	REC	W/A	BOPs (G)	Top-1
FP32	-	32/32	1858	72.54
HAWQ-V3† [41]		MP/MP	72	70.22
MSEinit*		6/6	65.31	71.61
I2LP (ours)*		MP/MP	65.30	**71.75**
MSEinit*		4/4	29.03	33.68
I2LP (ours)*		MP/MP	29.01	**55.50**
QDrop* [34]	✓	4/4	29.03	69.13
MRECG [23]	✓	4/4	29.03	69.46
recI2LP (ours)*	✓	MP/MP	28.79	**70.21**

(b) Results on MobileNet-V2

Accuracy vs. model size

Method	REC	W/A	MS (MiB)	Top-1
FP32	-	32/32	13.4	72.89
BRECQ	✓	MP/8	1.5	70.28
EMQ	✓	MP/8	1.5	70.75
recI2LP (ours)	✓	MP/8	1.4999	**71.99**
BRECQ	✓	MP/8	1.3	68.99
OMPQ	✓	MP/8	1.3	69.62
EMQ	✓	MP/8	1.3	70.72
recI2LP (ours)	✓	MP/8	1.29997	**71.02**

Accuracy vs. BOPs

Method	REC	W/A	BOPs (G)	Top-1
FP32	-	32/32	307	72.89
QDrop*† [34]	✓	4/4	5.39	68.52
recI2LP (ours)*	✓	MP/MP	5.37	**69.61**
QDrop* [34]	✓	4/4	4.812	66.77
recI2LP (ours)*	✓	MP/MP	4.806	**68.54**
MSEinit*		4/4	4.812	3.22
I2LP (ours)*		MP/MP	4.811	**53.53**

(c) Results on DeiT: Accuracy vs. BOPs

			DeiT-T		DeiT-S		DeiT-B	
Method	REC	W/A	BOPs (G)	Top-1	BOPs (G)	Top-1	BOPs (G)	Top-1
FP32	-	32/32	1284	72.13	4709	79.83	17791	81.80
RepQViT+*†		6/6	45.95	71.01	167.19	78.91	635.6	79.38
I2LP (ours)*		MP/MP	45.93	**71.32**	167.16	**79.05**	634.3	**81.26**
RepQViT+*		6/6	45.13	71.06	165.56	78.73	632.30	80.09
I2LP (ours)*		MP/MP	45.06	**71.26**	165.25	**79.07**	632.25	**81.37**
RepQViT+*†		4/4	21.46	59.20	76.38	69.28	286.6	75.71
I2LP (ours)*		MP/MP	21.31	**62.69**	76.34	**73.29**	286.4	**76.79**
RepQViT+*		4/4	20.05	56.76	73.58	68.12	281.0	74.57
I2LP (ours)*		MP/MP	20.02	**59.63**	73.37	**72.70**	280.8	**75.43**
RepQViT+*†		3/3	12.88	9.03	44.59	21.65	166.47	46.46
I2LP (ours)*		MP/MP	12.87	**21.48**	44.56	**37.90**	164.43	**50.94**

Table 2: Comparison with LAMPQ and uniform AdaLog on DeiT under parameter-weighted average-bit budgets. Top-1 accuracy (%). * denotes our implementation.

Budget	Method	DeiT-T	DeiT-S	DeiT-B
4-bit	AdaLog*	63.44	72.65	78.35
	LAMPQ	65.71	75.40	79.24
	I2LP (ours)*	**66.22**	**75.89**	**79.64**
3-bit	AdaLog*	32.97	31.10	59.44
	LAMPQ	37.54	45.38	61.44
	I2LP (ours)*	**42.09**	**53.62**	**69.07**

Table 3: Comparison of uniform- and mixed-precision PTQ methods for object detection using DETR-R50 on COCO 2017.

Method	W/A	BOPs (T)	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
FP32	32/32	159.9	41.9	62.3	44.1	20.3	45.8	60.9
MSEinit*	8/8	9.9943	39.4	61.1	41.5	17.6	43.1	58.7
I2LP (ours)*	MP/MP	9.9942	**41.4**	**61.9**	**43.5**	**20.0**	**45.4**	**60.7**
MSEinit*	6/6	5.6217	17.0	41.0	11.9	2.1	13.6	34.6
I2LP (ours)*	MP/MP	5.6215	**38.6**	**58.4**	**40.5**	**16.7**	**42.4**	**58.2**
MSEinit*	5/5	3.9040	1.5	7.4	0.2	0.1	0.5	4.4
I2LP (ours)*	MP/MP	3.9039	**33.7**	**53.6**	**34.2**	**12.4**	**36.9**	**53.6**

Table 4: DINOv3 results under BOPs budgets matched to uniform 6-bit quantization. We report ImageNet-1K linear-probe Top-1 accuracy (%). U6 denotes uniform 6-bit quantization, and gains are measured in percentage points.

Model	BOPs U6 (T)	BOPs I2LP (T)	U6 Top-1	I2LP Top-1	Gain (pp)
DINOv3-S	0.1691	0.1688	63.24	**69.41**	+6.17
DINOv3-B	0.6455	0.6443	64.56	**77.15**	+12.59
DINOv3-L	2.2620	2.2570	78.01	**84.08**	+6.07
DINOv3-H+	6.1960	6.1910	81.98	**87.89**	+5.91

of accepted iterations. Under the 4-bit BOPs constraint, the task loss decreases monotonically over accepted iterations and test accuracy improves accordingly. The inset shows that, at the third iteration, $M = 4$ yields the lowest task loss despite larger M achieving lower surrogate loss, supporting the use of multiple update-size candidates.

Search policies and runtime modes. We compare search policies under the same 4-bit BOPs budget. For iterative policies, refinement stops when the evaluated task loss no longer decreases. The $\mathcal{U} = \{1\}$ policy performs strictly local one-layer ILP updates. Since this policy cannot make a large resource-reducing move from a high-precision reference, we initialize it from the feasible uniform 4-bit configuration. The $\mathcal{U} = \{N\}$ policy solves only the no-update-size-limit ILP at each refinement step. We also compare against a greedy resource-reduction baseline that lowers one layer at a time according to the ratio between surrogate-loss increase and resource reduction. The full I2LP policy solves one ILP for each $M \in \mathcal{U}_{\text{multi}}$, where $\mathcal{U}_{\text{multi}} = \{2, 4, \ldots, 2^{\lfloor \log_2 N \rfloor}, N\}$, and selects the candidate with the lowest evaluated task loss.

Table 5(a) shows that I2LP outperforms these simpler search policies. The poor performance of the $\mathcal{U} = \{1\}$ policy indicates that overly local updates are insufficient, even when initialized from a feasible 4-bit reference. Under a tight resource budget, increasing the bit-width of one layer may require decreasing that of another, which cannot be achieved in a single-layer update. The gap between the $\mathcal{U} = \{N\}$ policy and I2LP further shows the benefit of evaluating multiple update-size-constrained candidates rather than relying on a single no-limit surrogate optimum.

A separate exhaustive-search analysis on small ResNet-18 subproblems further shows that I2LP refinement reduces the average gap to the exact task-loss optimum compared with the initial one-shot ILP solution (see Appendix I).

We also investigate which update-size limits are selected during refinement. Across 41 tested settings covering ResNet-18, MobileNet-V2, DeiT, and DETR under different BOPs and memory budgets, $M = N$ is never selected after the first iteration. Moreover, about 90% of accepted refinement steps use $M \in \{2, 4, 8\}$, indicating that large-update candidates mainly serve as early exploration options, while later refinement is dominated by small-update regimes.

Table 5(b) reports representative runtime modes. The parentheses indicate the step-size/zero-point calibration mode used during bit selection. I2LP (DR) uses dynamic-rough calibration during bit selection. I2LP (Static) + QDrop uses static calibration during bit selection and applies QDrop only after the bit configuration is fixed. recI2LP-200 (DR) additionally performs QDrop reconstruction dur-

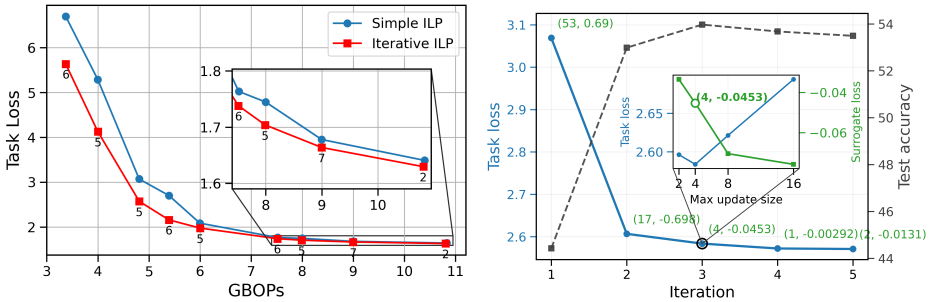


Figure 4: Behavior of iterative refinement on MobileNet-V2. (a) Task loss (KL divergence) on calibration data across BOPs budgets, comparing the no-update-size-limit ILP and the proposed I2LP. Numbers under the I2LP curve indicate the number of accepted iterations until convergence. (b) Main panel: task loss and test accuracy over iterations under the 4-bit BOPs constraint. Green annotations report (the actual number of changed layers, surrogate loss) for the selected candidate at each iteration. Inset: candidate task loss and surrogate loss at the third iteration for different update-size limits M.

Table 5: Search-policy and runtime ablations on ResNet-18 under the 4-bit BOPs constraint.

(a) Search-policy ablation.

Policy	Top-1 (%)
I2LP, $\mathcal{U} = \{1\}$ (local)	37.81
Greedy resource reduction	48.91
I2LP, $\mathcal{U} = \{N\}$ (no-limit only)	52.66
I2LP, $\mathcal{U}_{\text{multi}}$	**55.50**

(b) Representative runtime modes. DR denotes dynamic-rough calibration during bit selection.

Mode	Time	Top-1 (%)
I2LP (DR)	13m	55.50
I2LP (Static) + QDrop	20m	69.58
recI2LP-200 (DR)	1h44m	**70.00**

ing surrogate construction. These modes make the accuracy–runtime trade-off explicit. Appendix H reports additional runtime results and ILP solver statistics, confirming that solver time is negligible compared with surrogate construction and candidate evaluation. In a separate matched-budget comparison with our reproduction of OMPQ, I2LP achieves higher accuracy with shorter search time (see Appendix H).

Effect of reconstruction-aware bit selection. Reconstruction-aware bit selection aligns the search objective with the post-reconstruction model. As shown in Table 6(a), recI2LP-200 achieves lower accuracy than I2LP before reconstruction (46.54% vs. 55.50%), but higher accuracy afterward (70.00% vs. 69.58%). Table 6(b) further shows that using 1,000 QDrop steps during surrogate construction achieves 70.21%, close to 70.24% with 5,000 steps. Additional discussion is provided in Appendix G.

Table 6: Ablation on reconstruction-aware bit selection under the 4-bit BOPs constraint on ResNet-18.

(a) I2LP vs. recI2LP.

Method	after MSE init.	after recon.
I2LP	**55.50**	69.58
recI2LP-200	46.54	**70.00**

(b) QDrop recon. steps for $\Delta\mathcal{L}$ (final QDrop steps fixed to 20k).

Iters	20	200	1000	5000
recI2LP	69.87	70.00	70.21	70.24

5 Conclusion

We introduced I2LP, an iterative ILP-based bit-selection method that limits the update size to reduce the mismatch between the surrogate loss and the evaluated task loss, yielding better mixed-precision configurations. We also presented a reconstruction-aware variant (recI2LP) that aligns the surrogate with the post-reconstruction objective in PTQ. Across PTQ and QAT on CNNs, ViTs, and DETR under diverse model-size and BOP constraints, our methods consistently improve the accuracy–efficiency trade-off over strong uniform-precision and mixed-precision baselines. We further demonstrate the applicability of I2LP to weight-only quantization in a large language model.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge the use of ABCI 3.0, provided by AIST and AIST Solutions, and the TSUBAME4.0 supercomputer at Institute of Science Tokyo.

Funding. This work was supported by DENSO IT Laboratory.

Competing interests. The authors declare no competing interests.

References

[1] Mehmet Emre Akbulut, Hazem Hesham Yousef Shalby, Fabrizio Pittorino, and Manuel Roveri. Infoq: Mixed-precision quantization via global information flow. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, pages 19598–19606, 2026.

[2] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 696–697, 2020.

[3] Arun Chauhan, Utsav Tiwari, et al. Post training mixed precision quantization of neural networks using first-order information. In Proceedings of the IEEE/CVF international conference on computer vision, pages 1343–1352, 2023.

[4] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. International Conference on Learning Representation, 2018.

[5] 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, pages 248–255. Ieee, 2009.

[6] Peijie Dong, Lujun Li, Zimian Wei, Xin Niu, Zhiliang Tian, and Hengyue Pan. Emq: Evolving training-free proxies for automated mixed precision quantization. In Proceedings of the IEEE/CVF international conference on computer vision, pages 17076–17086, 2023.

[7] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In Proceedings of the IEEE/CVF international conference on computer vision, pages 293–302, 2019.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2020.

[9] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. In International Conference on Learning Representations, 2019.

[10] Shinya Gongyo, Jinrong Liang, Mitsuru Ambai, Rei Kawakami, and Ikuro Sato. Learning non-uniform step sizes for neural network quantization. In Proceedings of the Asian Conference on Computer Vision, pages 4385–4402, 2024.

[11] Seokho Han, Seoyeon Yoon, Jinhee Kim, Dongwei Wang, Kang Eun Jeon, Huanrui Yang, and Jong Hwan Ko. Msq: Memory-efficient bit sparsification quantization. In 2025 IEEE/CVF International Conference on Computer Vision (ICCV), pages 21885–21894. IEEE, 2025.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016.

[13] Xijie Huang, Zhiqiang Shen, Shichao Li, Zechun Liu, Hu Xianghong, Jeffry Wicaksana, Eric Xing, and Kwang-Ting Cheng. Sdq: Stochastic differentiable quantization with mixed precision. In International Conference on Machine Learning, pages 9295–9309. PMLR, 2022.

[14] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Accurate post training quantization with small calibration sets. In International conference on machine learning, pages 4466–4475. PMLR, 2021.

[15] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2704–2713, 2018.

[16] Han-Byul Kim, Joo Hyung Lee, Sungjoo Yoo, and Hong-Seok Kim. Metamix: meta-state precision searcher for mixed-precision activation quantization. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pages 13132–13141, 2024.

[17] Minjun Kim, Jaeri Lee, Jongjin Kim, Jeongin Yun, Yongmo Kwon, and U Kang. Lampq: Towards accurate layer-wise mixed precision quantization for vision transformers. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, pages 5691–5699, 2026.

[18] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In ICLR, February 2021.

[19] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. RepQ-ViT: Scale reparameterization for post-training quantization of vision transformers. In ICCV, pages 17227–17236, October 2023.

[20] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. Proceedings of machine learning and systems, 6:87–100, 2024.

[21] Zechun Liu, Kwang-Ting Cheng, Dong Huang, Eric P Xing, and Zhiqiang Shen. Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4942–4952, 2022.

[22] Yuexiao Ma, Taisong Jin, Xiawu Zheng, Yan Wang, Huixia Li, Yongjian Wu, Guannan Jiang, Wei Zhang, and Rongrong Ji. Ompq: Orthogonal mixed precision quantization. In Proceedings of the AAAI conference on artificial intelligence, volume 37, pages 9029–9037, 2023.

[23] Yuexiao Ma, Huixia Li, Xiawu Zheng, Xuefeng Xiao, Rui Wang, Shilei Wen, Xin Pan, Fei Chao, and Rongrong Ji. Solving oscillation problem in post-training quantization through a theoretical perspective. In CVPR, pages 7950–7959, 2023. doi: 10.1109/CVPR52729.2023.00768.

[24] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. 2020.

[25] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In Proceedings of the 39th International Conference on Machine Learning, pages 16318–16330, 2022. URL `https://proceedings.mlr.press/v162/nagel22a.html`.

[26] Eunhyeok Park and Sungjoo Yoo. Profit: A novel training method for sub-4-bit mobilenet models. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16, pages 430–446. Springer, 2020.

[27] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. 28, 2015.

[29] Juncheol Shin, Junhyuk So, Sein Park, Seungyeop Kang, Sungjoo Yoo, and Eunhyeok Park. Nipq: Noise proxy-based integrated pseudo-quantization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 3852–3861, 2023.

[30] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. arXiv preprint arXiv:2508.10104, 2025.

[31] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, pages 6105–6114. PMLR, 2019.

[32] Stefan Uhlich, Lukas Mauch, Fabien Cardinaux, Kazuki Yoshiyama, Javier Alonso Garcia, Stephen Tiedemann, Thomas Kemp, and Akira Nakamura. Mixed precision dnns: All you need is a good parametrization. arXiv preprint arXiv:1905.11452, 2019.

[33] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8612–8620, 2019.

[34] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. QDrop: Randomly dropping quantization for extremely low-bit post-training quantization. In ICLR, 2022.

[35] Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Vajda, and Kurt Keutzer. Mixed precision quantization of convnets via differentiable neural architecture search. arXiv preprint arXiv:1812.00090, 2018.

[36] Zhuguanyu Wu, Jiaxin Chen, Hanwen Zhong, Di Huang, and Yunhong Wang. Adalog: Post-training quantization for vision transformers with adaptive logarithm quantizer. In European Conference on Computer Vision, pages 411–427. Springer, 2024.

[37] Zhuguanyu Wu, Shihe Wang, Jiayi Zhang, Jiaxin Chen, and Yunhong Wang. Fima-q: Post-training quantization for vision transformers by fisher information matrix approximation. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 14891–14900, 2025.

[38] Zhuguanyu Wu, Jiayi Zhang, Jiaxin Chen, Jinyang Guo, Di Huang, and Yunhong Wang. Aphq-vit: Post-training quantization with average perturbation hessian based reconstruction for vision transformers. In 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9686–9695. IEEE, 2025.

[39] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025.

[40] Linjie Yang and Qing Jin. Fracbits: Mixed precision quantization via fractional bit-widths. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 10612–10620, 2021.

[41] Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael Mahoney, et al. Hawq-v3: Dyadic neural network quantization. In International Conference on Machine Learning, pages 11875–11886. PMLR, 2021.

[42] Zhaoyang Zhang, Wenqi Shao, Jinwei Gu, Xiaogang Wang, and Ping Luo. Differentiable dynamic quantization with mixed precision and adaptive resolution. In International Conference on Machine Learning, pages 12546–12556. PMLR, 2021.

A Sensitivity of quantization on MobileNet-V2

We visualize the quantization sensitivity in shallow-to-deep order, MACs order, and memory order in MobileNet-V2 in Fig.5. We assign 8 bit in all layers and change only a single layer to 3 bit. We can see that the resulting accuracy degradation differs significantly between layers.

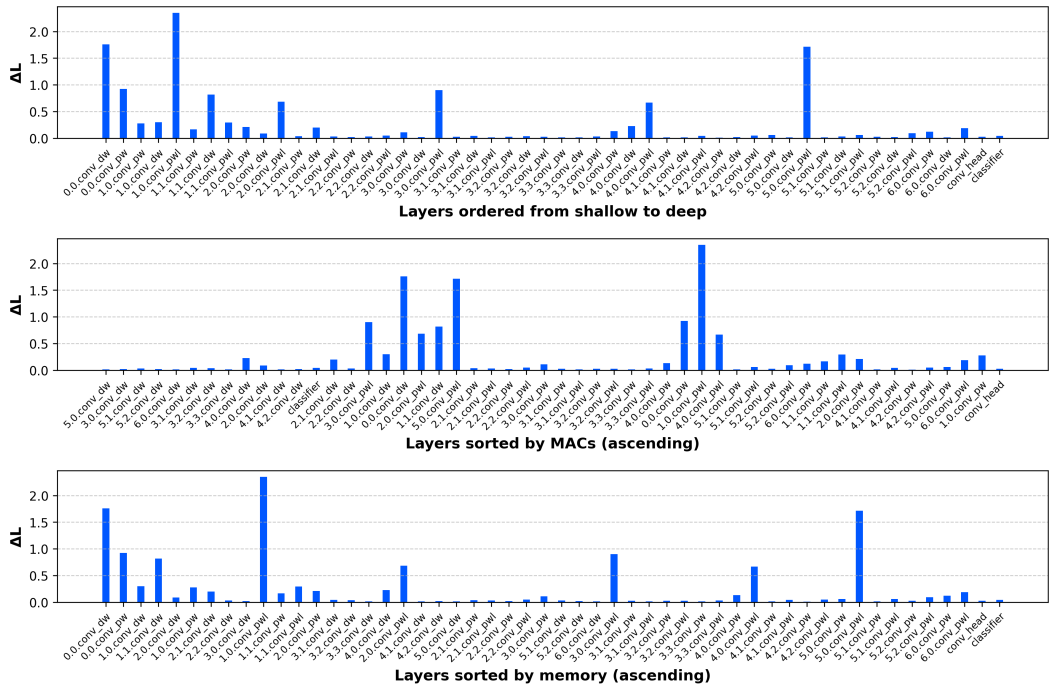


Figure 5: Sensitivity of quantization on each layer on MobileNet-V2. Each layer is quantized to 3 bit, while others are set to 8 bit.

B Theoretical Studies on Sufficient Conditions for Surrogate Agreement

Let $\mathcal{F} = \{\mathbf{b} \in \mathcal{B}^N \mid C(\mathbf{b}) \leq C_{\max}, |S(\mathbf{b})| \leq M\}$ denote the nonempty feasible set of Eq. (2). We provide simple sufficient conditions under which the additive surrogate preserves the task-loss minimizer over a fixed feasible set. This analysis is intended to clarify the role of interaction errors; I2LP itself does not rely on these assumptions.

For simplicity, we use $\mathcal{L}(\mathbf{b})$ and $\Delta\mathcal{L}(b_i)$ instead of $L_{\text{task}}(\mathbf{b})$ and $\Delta L_i(b_i)$, respectively, throughout this appendix. The original combinatorial problem in (1) and additive surrogate formulation in (2) are related as follows.

Theorem B.1 *Let $\delta > 0$ be a fixed constant and let $\mathbf{b}^*$ be the optimal solution for the ILP in* (3)*. Suppose the following two conditions hold for all $\mathbf{b} \in \mathcal{F}$:*

$$\left| \mathcal{L}(\mathbf{b}) - \sum_{i=1}^{N} \mathcal{L}(\mathbf{b}_i^0) - \left(\mathcal{L}(\mathbf{b}^*) - \sum_{i=1}^{N} \mathcal{L}(\mathbf{b}_i^{*0}) \right) \right| \leq \delta,$$

$$\sum_{i=1}^{N} \Delta\mathcal{L}(b_i) \geq \min_{\mathbf{b} \in \mathcal{F}} \sum_{i=1}^{N} \Delta\mathcal{L}(b_i) + 2\delta \quad \textit{for} \quad \mathbf{b} \neq \mathbf{b}^*.$$

Then the minimizers of the original and ILP losses coincide:

$$\mathbf{b}^* = \arg\min_{\mathbf{b} \in \mathcal{F}} \mathcal{L}(\mathbf{b}). \tag{4}$$

From this theorem the following two corollaries can be derived, which are more intuitive.

Corollary B.2 (Independent losses) *Under the assumption that the bit-width of each substructure affects only its own loss contribution, i.e., the loss decomposes as $\mathcal{L}(\mathbf{b}) = \sum_i L_i(b_i)$ with the loss functions per substructure $L_i(b_i)$ then $\min_{\mathbf{b} \in \mathcal{F}} \mathcal{L}(\mathbf{b}) = \min_{\mathbf{b} \in \mathcal{F}} \left[\sum_{i=1}^{N} \Delta\mathcal{L}(b_i) + \mathcal{L}(\mathbf{b^0}) \right]$, and* (4) *holds.*

This suggests that weak inter-substructure interactions reduce the surrogate–task mismatch, although agreement of the minimizers also depends on the surrogate gap. In contrast, strong correlations often lead to substantial mismatches. As described below, this effect is particularly pronounced in reconstruction-aware bit selection. The next corollary suggests another strategy to reduce such mismatches.

Corollary B.3 (Local neighborhood) *Under the assumption that the search space $\mathcal{F}$ is restricted to configurations that differ from the reference $\mathbf{b}^0$ in at most one substructure, i.e., satisfying $d_H(\mathbf{b}, \mathbf{b}^0) \leq 1$, where d_H denotes the Hamming distance over substructures,* (4) *holds.*

Here, $d_H(\mathbf{b}, \mathbf{b}^0)$ is defined regardless of the magnitude of the bit-width change (e.g., 8→2 and 8→7 both contribute one to the distance).

Corollary B.3 shows that (4) is guaranteed when the bit-width configuration is updated by only a single substructure (i.e., within a Hamming distance of 1). However, such strictly local updates are inefficient: they require many iterations to make progress and are prone to being trapped in poor local minima. This situation is analogous to gradient descent with a very small learning rate—loss reduction is guaranteed, but convergence is slow and the risk of stagnation in local minima is high. By analogy, just as gradient descent benefits from a moderately large learning rate, our framework favors using a larger update size M to accelerate progress and escape poor local solutions.

B.1 Relation to Hamming Trust-Region Search

The constraint $|S(\mathbf{b})| \leq M$ can be viewed as a Hamming trust region, where M is a discrete radius around the current reference. I2LP is not a fixed-radius local search: it solves ILPs for multiple radii $M \in \mathcal{U}^{(t)}$, evaluates the candidates by task loss, and selects the effective radius adaptively. This matters because the surrogate objective is monotonic in M, whereas the task loss can be non-monotonic (Figure 1); Table 5 shows that the multi-M policy outperforms both $\mathcal{U} = \{1\}$ and $\mathcal{U} = \{N\}$.

B.2 Proof of Theorem B.1

The assumptions of Theorem B.1 are

$$\left| \mathcal{L}(\mathbf{b}) - \sum_{i=1}^{N} \mathcal{L}(\mathbf{b}_i^0) - \left(\mathcal{L}(\mathbf{b}^*) - \sum_{i=1}^{N} \mathcal{L}(\mathbf{b}_i^{*0}) \right) \right| \leq \delta, \tag{5}$$

$$\sum_{i=1}^{N} \Delta\mathcal{L}(b_i) \geq \min_{\mathbf{b} \in \mathcal{F}} \sum_{i=1}^{N} \Delta\mathcal{L}(b_i) + 2\delta \quad \text{for} \quad \mathbf{b} \neq \mathbf{b}^*. \tag{6}$$

We define $f(\mathbf{b})$ as

$$f(\mathbf{b}) := \sum_{i=1}^{N} \Delta\mathcal{L}(b_i), \qquad \Delta\mathcal{L}(b_i) = \mathcal{L}(\mathbf{b}_i^0) - \mathcal{L}(\mathbf{b}^0). \tag{7}$$

Let $\mathbf{b}^* \in \mathcal{F}$ be an optimal solution of the ILP that minimizes f over $\mathcal{F}$. Define the interaction (non-additivity) error

$$e(\mathbf{b}) := \big(\mathcal{L}(\mathbf{b}) - \mathcal{L}(\mathbf{b}^0)\big) - f(\mathbf{b}). \tag{8}$$

Using the notation, the first condition is expressed as

$$\begin{aligned} &|e(\mathbf{b}) - e(\mathbf{b}^*)| \\ &= \left| \mathcal{L}(\mathbf{b}) - \sum_{i=1}^{N} \mathcal{L}(\mathbf{b}_i^0) - \left(\mathcal{L}(\mathbf{b}^*) - \sum_{i=1}^{N} \mathcal{L}(\mathbf{b}_i^{*0}) \right) \right| \leq \delta. \end{aligned} \tag{9}$$

For $\delta > 0$ we have following conditions for all $\mathbf{b} \in \mathcal{F}$:

$$\big|e(\mathbf{b}) - e(\mathbf{b}^*)\big| \leq \delta, \tag{10}$$

$$f(\mathbf{b}) \geq f(\mathbf{b}^*) + 2\delta \quad \text{for all } \mathbf{b} \in \mathcal{F} \setminus \{\mathbf{b}^*\}. \tag{11}$$

By using

$$\mathcal{L}(\mathbf{b}) - \mathcal{L}(\mathbf{b}^*) = \big[f(\mathbf{b}) - f(\mathbf{b}^*)\big] + \big[e(\mathbf{b}) - e(\mathbf{b}^*)\big],$$

$f(\mathbf{b}) - f(\mathbf{b}^*) \geq 2\delta$, and (10), $e(\mathbf{b}) - e(\mathbf{b}^*) \geq -\delta$,

$$\mathcal{L}(\mathbf{b}) - \mathcal{L}(\mathbf{b}^*) \;\geq\; 2\delta + (-\delta) \;=\; \delta \;>\; 0.$$

Therefore $\mathcal{L}(\mathbf{b}) > \mathcal{L}(\mathbf{b}^*)$ for all $\mathbf{b} \neq \mathbf{b}^*$, proving that $\mathbf{b}^*$ is the minimizer of $\mathcal{L}$.

B.3 Proof of Corollary B.2

When all substructures are independent, $\mathcal{L}(\mathbf{b}) = \sum_i L_i(b_i)$ is satisfied. This leads to

$$\begin{aligned}
&\mathcal{L}(\mathbf{b}) - \sum_{i=1}^{N} \mathcal{L}(\mathbf{b}_i^0) - \left(\mathcal{L}(\mathbf{b}^*) - \sum_{i=1}^{N} \mathcal{L}(\mathbf{b}_i^{*0})\right) \\
&= \sum_i L_i(b_i) - \sum_i \left(L_i(b_i) + \sum_{j \neq i} L_j(b_j^0)\right) \\
&- \left(\sum_i L_i(b_i^*) - \sum_i \left(L_i(b_i^*) + \sum_{j \neq i} L_j(b_j^0)\right)\right) \\
&= 0. \qquad (12)
\end{aligned}$$

Therefore, the surrogate and task losses differ only by a constant, and Eq. (4) holds.

B.4 Proof of Corollary B.3

We assume that the search space $\mathcal{F}$ is restricted to configurations satisfying $d_H(\mathbf{b}, \mathbf{b}^0) \leq 1$. Define the search space as $\mathcal{F}_0$. In this special case, all $\mathbf{b} \in \mathcal{F}_0$ are expressed as $\mathbf{b}_j^0$ for some j. By definition,

$$b^* = \arg\min_{\mathbf{b} \in \mathcal{F}_0} \sum_{i=1}^{N} \Delta\mathcal{L}(b_i) = \arg\min_{\mathbf{b} \in \mathcal{F}_0} \mathcal{L}(\mathbf{b}) \qquad (13)$$

is satisfied. Equivalently, for any $\mathbf{b} \in \mathcal{F}_0$,

$$\mathcal{L}(\mathbf{b}) - \sum_{i=1}^{N} \mathcal{L}(\mathbf{b}_i^0) = -(N-1)\mathcal{L}(\mathbf{b}^0), \qquad (14)$$

and

$$\left|\mathcal{L}(\mathbf{b}) - \sum_{i=1}^{N} \mathcal{L}(\mathbf{b}_i^0) - \left(\mathcal{L}(\mathbf{b}^*) - \sum_{i=1}^{N} \mathcal{L}(\mathbf{b}_i^{*0})\right)\right| = 0. \qquad (15)$$

Under the assumption that the search space is restricted to configurations that differ from the reference $\mathbf{b}^0$ in at most one substructure, i.e., $d_H(\mathbf{b}, \mathbf{b}^0) \leq 1$, Eq. (4) holds.

C Additional Experimental Details

Calibration data and task loss. For bit selection, we use 1024 unlabeled calibration samples for ImageNet classification and 64 unlabeled calibration samples for COCO object detection. No data augmentation is used during bit selection. The task loss is evaluated as the distillation loss between the full-precision and quantized networks with temperature 1. For QAT experiments, bit selection is performed once using unlabeled calibration data, and the selected configuration is then trained with the full labeled dataset.

Candidate bit-widths and initialization. We perform per-layer bit-width selection. For image classification, the initial reference configuration is uniform 8-bit and the candidate set is $\mathcal{B} = \{2, 3, 4, 5, 6, 7, 8\}$. For object detection, the initial reference configuration is uniform 16-bit and the candidate set is $\mathcal{B} = \{2, 4, 8, 16\}$. Unless otherwise specified, I2LP uses $\mathcal{U} = \{N\}$ in the first iteration to avoid infeasible ILP solutions, and then uses

$$\mathcal{U} = \{2, 4, \ldots, 2^{\lfloor \log_2 N \rfloor}, N\}$$

in later iterations.

BOPs computation. For ImageNet classification, the standard preprocessing uses a fixed input resolution, so the resulting BOPs do not vary across batches. For COCO with DETR, images are resized with short side 800 and maximum size 1333, and then padded to a common shape within each batch. Since this can make BOPs batch-dependent, we use a single fixed calibration batch of 64 images and report BOPs computed on this batch.

Uniform MSE-initialized counterparts. In the main tables, MSEinit denotes a uniform-bit counterpart that uses the same calibration data, MSE-based step-size/zero-point initialization, and evaluation protocol as vanilla I2LP. Thus, MSEinit serves as a controlled uniform-bit counterpart, and the comparison between MSEinit and I2LP isolates the effect of changing the bit configuration.

QAT protocol. For QAT experiments, we follow the experimental protocol of MetaMix [16] for training after bit selection. The selected bit configuration is fixed during QAT, and LSQ [9] is used for quantized training. We use the same teacher models as MetaMix: ResNet-101 for ResNet-18 and MobileNet-V2 with a $1.2\times$ width multiplier for MobileNet-V2.

D Analysis for ΔL in reconstruction-aware I2LP

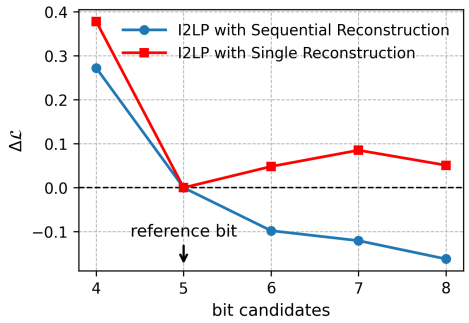


Figure 6: Task loss change across bit candidates with 5-bit reference point in the first block of MobileNet-V2.

I2LP can integrate with the reconstruction methods, as outlined in Algorithm 1. In conventional reconstruction methods [18, 34], reconstruction proceeds sequentially from the first block to the last. As a result, later blocks are strongly influenced by earlier ones, leading to strong correlations between blocks, which is far from the independent assumption in Corollary B.2. This implies that naively estimating the loss variation by changing the bit-width of a single block and then minimizing its reconstruction error introduces a significant mismatch between true and estimated variations. Indeed, as shown in Figure 6 (red line), even when the bit-width of a block is increased and its representational capacity should improve, the task loss increases, yielding $\Delta\mathcal{L} > 0$. Consequently, minimizing the reconstruction error only for the changed block does not provide a reliable estimate of the true change in task loss.

To address this issue, we optimize the reconstruction error sequentially for all blocks following the changed block. This adjustment ensures that the estimate more faithfully reflects the true change in task loss. Figure 6 (blue line) shows that increasing the bit-width of a block results in $\Delta\mathcal{L} < 0$, consistent with the expected behavior.

E Surrogate computation for I2LP/recI2LP

The surrogate terms in Algorithm 1 can be computed using Algorithm 2. As discussed in the main text, the evaluations over substructures i and candidate bits $n^{(k)}$ are embarrassingly parallel.

Recalibration of step size and zero point. When evaluating a candidate bit assignment, the quantization step size and zero point must be calibrated for the corresponding bit-width. We consider three calibration modes during bit selection: (i) static, (ii) dynamic-rough, and (iii) dynamic-fine. These modes provide an accuracy–runtime trade-off and are used in the runtime comparison in Appendix H.

Algorithm 2 ComputeSurrogate: computing $\{\Delta\mathcal{L}_{i,k}\}$ at reference $\mathbf{b}^0$

Input: reference $\mathbf{b}^0$; candidate bits $\mathcal{B}$; calibration data $\mathcal{D}$; calibration mode $\in \{\text{STATIC}, \text{DYNAMIC-ROUGH}, \text{DYNAMIC-FINE}\}$; mode $\in \{\text{VANILLA}, \text{RECONSTRUCTION}\}$
Output: surrogate terms $\{\Delta\mathcal{L}_{i,k}\}_{i=1..N,\, k=1..K}$
Evaluate $\mathcal{L}(\mathbf{b}^0)$ on $\mathcal{D}$ once
for each layer $i = 1$ **to** N **do**
  for each bit $n^{(k)} \in \mathcal{B}$ **do**
    $\mathbf{b}^0_{\hat{i}} \leftarrow [b^0_1, \ldots, b^0_{i-1}, n^{(k)}, b^0_{i+1}, \ldots, b^0_N]$
    Calibrate the step size and zero point of layer i under $\mathbf{b}^0_{\hat{i}}$ using the specified calibration mode
    if mode = RECONSTRUCTION **then**
      RECONSTRUCT$(i, \mathbf{b}^0_{\hat{i}}, \mathcal{D})$ {Alg. 3}
    end if
    Evaluate $\mathcal{L}(\mathbf{b}^0_{\hat{i}})$ on $\mathcal{D}$
    $\Delta\mathcal{L}_{i,k} \leftarrow \mathcal{L}(\mathbf{b}^0_{\hat{i}}) - \mathcal{L}(\mathbf{b}^0)$
  end for
end for
return $\{\Delta\mathcal{L}_{i,k}\}$

(i) Static calibration. In the static mode, we precompute the step size and zero point before bit selection for each candidate bit-width using the corresponding uniform-precision configuration. During surrogate evaluation, when only substructure i is changed to bit-width b_i, we simply reuse the precomputed step size and zero point for that bit-width. This is the cheapest option because it avoids recalibrating quantization parameters for each modified substructure.

(ii) Dynamic-rough calibration. In the dynamic-rough mode, we recalibrate the step size and zero point for the modified substructure during surrogate evaluation. Following the common MSE-based calibration used in PTQ methods such as QDrop [34], we perform a lightweight grid search over percentile-based clipping ranges and choose the quantization parameters that minimize the MSE between the original and quantized tensors. This mode is more expensive than static calibration but better adapts the quantization parameters to the current modified configuration.

(iii) Dynamic-fine calibration. In the dynamic-fine mode, we further refine the quantization parameters after dynamic-rough calibration. Starting from the rough estimate, we alternately update the step size and zero point to minimize the MSE. During this refinement, we alternate between fixing the quantized integer assignments and optimizing the dequantization scale and offset, and then updating the integer assignments using the updated parameters. This alternating procedure further reduces the MSE while keeping the optimization stable. This mode provides a more accurate MSE calibration but is the most expensive among the three modes.

For activation quantizers, the MSE-optimal step size and zero point can vary across calibration mini-batches. To reduce this batch dependence, we use a moving average of the activation quantization parameters over calibration batches. If (s_t, z_t) denotes the step size and zero point estimated from the t-th calibration batch, we update

$$\bar{s}_t = \rho\bar{s}_{t-1} + (1-\rho)s_t, \qquad \bar{z}_t = \rho\bar{z}_{t-1} + (1-\rho)z_t,$$

and use the running values $(\bar{s}_t, \bar{z}_t)$ when evaluating $\Delta\mathcal{L}_{i,k}$. This stabilizes the surrogate terms by reducing their dependence on a particular calibration mini-batch. We use $\rho = 0.9$ in all experiments.

Unless otherwise specified, we use dynamic-rough calibration for vanilla I2LP and recI2LP. For the I2LP+QDrop runtime setting, we use static calibration during bit selection and apply QDrop only after the selected bit configuration is fixed.

Computation of ΔL in recI2LP. In recI2LP, $\Delta\mathcal{L}_{i,k}$ is computed after reconstruction because the post-reconstruction loss is the target objective. To account for cross-block dependencies, we reconstruct sequentially from the modified block to the end.

Algorithm 3 RECONSTRUCT$(i, \mathbf{b}, \mathcal{D})$: reconstruction for evaluating $\Delta\mathcal{L}_{i,k}$ (recI2LP)

Input: configuration $\mathbf{b}$; calibration data $\mathcal{D}$; reconstruction steps T
Output: reconstructed model parameters (round parameters for Adaround-based methods)
Identify the reconstruction block m_i that includes the layer with index i
for each block $\ell = m_i$ **to** last block **do**
    Run block-wise reconstruction for ℓ under $\mathbf{b}$ for T steps (e.g., QDrop)
end for
return reconstructed network

The calibration mode and whether reconstruction is used during surrogate computation determine the dominant term $T_\Delta^{(t)}$ in the runtime decomposition below. Table 8 reports the corresponding accuracy–runtime modes used in our experiments.

F Additional Discussion on Update-Size Control

The update-size constraint $|S(\mathbf{b})| \leq M$ can be viewed as a Hamming-radius constraint around the current reference configuration, or as a discrete trust-region-like restriction. However, the role of M in I2LP is not to define a fixed local-search radius. At each refinement step, I2LP solves multiple ILPs with different update-size limits $M \in \mathcal{U}^{(t)}$, evaluates the resulting candidates by the evaluated task loss, and thereby selects the effective update size.

This effective update size is adaptive across refinement steps. Figure 4(b) shows that the selected update size can change over iterations: the best update size depends on the current reference and the stage of refinement. Thus, I2LP is not simply a fixed-radius Hamming local search. Rather, it repeatedly re-estimates the surrogate around the updated reference and uses the evaluated task loss to adapt the update size at each step.

The search-policy ablation in Table 5 supports this view: both the strictly local $\mathcal{U} = \{1\}$ policy and the no-limit $\mathcal{U} = \{N\}$ policy are worse than the multi-M policy.

G Additional Discussion on Reconstruction-Aware Bit Selection

Table 6 in the main text provides ablations for reconstruction-aware bit selection on ResNet-18 under the 4-bit BOPs constraint. Vanilla I2LP better optimizes the post-MSE-initialization loss, whereas recI2LP-200 better aligns bit selection with the post-reconstruction objective. The QDrop-step ablation shows that using 1000 reconstruction steps during surrogate estimation already approaches the 5000-step result, supporting the accuracy–runtime trade-off used in our experiments.

H Runtime and Solver Details

Compute environment. Unless otherwise specified, the experiments were run on NVIDIA H200 GPUs with 141GB memory. Table 8 reports representative wall-clock time and GPU-hours for ResNet-18 under the 4-bit BOPs constraint, and Table 7 reports ILP solver statistics.

Runtime decomposition. The wall-clock time of I2LP can be decomposed as

$$T_{\text{total}} = \sum_{t=1}^{n_{\text{iter}}} \left(T_\Delta^{(t)} + \sum_{M \in \mathcal{U}} T_{\text{ILP}}^{(t)}(M) + |\mathcal{U}| T_{\text{eval}}^{(t)} \right),$$

where $T_\Delta^{(t)}$ is the cost of constructing the surrogate terms $\{\Delta\mathcal{L}_{i,k}\}$, $T_{\text{ILP}}^{(t)}(M)$ is the solver time for the ILP with update-size limit M, and $T_{\text{eval}}^{(t)}$ is the cost of evaluating one candidate by the task loss. For vanilla I2LP, $T_\Delta^{(t)}$ consists of NK single-substructure task-loss evaluations. For recI2LP, $T_\Delta^{(t)}$ additionally includes reconstruction used to estimate $\Delta\mathcal{L}_{i,k}$.

The surrogate construction and candidate evaluations are embarrassingly parallel over modified substructures and update-size candidates. Therefore, when n_g GPUs are available, the wall-clock

Table 7: Representative ILP solver statistics for the no-update-size-limit case ($M = N$).

Model	Searchable layers	Candidate bits	Binary vars	Constraints	Solve time (s)
ResNet-18	21	7	147	21 assign. + 1 budget	0.037
MobileNet-V2	53	7	371	53 assign. + 1 budget	0.0387
DETR-R50	150	4	600	150 assign. + 1 budget	0.11

Table 8: End-to-end bit-selection runtime on ResNet-18 under the 4-bit BOPs constraint. GPU-hours are computed as wall-clock time multiplied by the number of GPUs.

Method	Step-size/ZP calib. in selection	Recon. in selection	QDrop steps	GPUs	Wall-clock / GPU-hours
I2LP	Dynamic-rough	No	–	1	13m / 0.22
I2LP+QDrop	Static	No	–	1	20m / 0.33
recI2LP-200	Dynamic-rough	Yes	200	1	1h44m / 1.73
recI2LP-1000 (optional)	Dynamic-fine	Yes	1000	8	1h18m / 10.40

time of these evaluation terms can be reduced approximately by parallel execution. In contrast, the ILP solver time is already small in our experimental regime and is not the runtime bottleneck.

Although exact MIP solving is worst-case exponential, the ILP instances arising in our experiments are small and highly structured. As shown in Table 7, even the no-update-size-limit ILP for DETR-R50 with 600 binary variables is solved in about 0.1 seconds. This confirms that the practical runtime is dominated by surrogate construction and candidate evaluation rather than ILP solving.

The 8-GPU recI2LP setting is an optional accelerated configuration for the highest-accuracy regime and is not required by the method. Vanilla I2LP and I2LP+QDrop both run on a single GPU, and recI2LP can also be run on a single GPU by reducing the number of QDrop reconstruction steps used during surrogate estimation.

Matched Search-Time Comparison with OMPQ. We compare I2LP with OMPQ [22] using its official implementation on ResNet-18 with one NVIDIA H200 GPU and no reconstruction. Both methods use the same pretrained model and matched resource budget with 8-bit first/last layers and 4-bit remaining layers. We use static step-size/zero-point calibration for I2LP and evaluate it with varying the number of calibration images, including 64 images to match OMPQ's setting. Search time includes sensitivity or surrogate construction and bit-allocation search, but excludes model loading and final ImageNet evaluation. At 64 calibration images, I2LP achieves 8.77 percentage points higher Top-1 accuracy than OMPQ (61.80% vs. 53.03%), while reducing search time from 37 to 22 seconds.

Table 9: Matched-budget accuracy and search time on ResNet-18. All runs use one H200 GPU, without reconstruction.

Method	Calib. images	Time (s)	Top-1 (%)
OMPQ	64	37	53.03
I2LP (Static)	32	13	61.09
I2LP (Static)	64	22	61.80
I2LP (Static)	128	49	61.12
I2LP (Static)	256	82	62.05
I2LP (Static)	512	147	62.08
I2LP (Static)	1,024	350	62.50

I Comparison with Exact Optima on Small Subproblems

We perform exhaustive searches over budget-feasible configurations in nine ResNet-18 subproblems with 8–16 selectable layers. We select the first N quantized layers in model order, allowing each

Table 10: Comparison with exact task-loss optima on small ResNet-18 subproblems under uniform-4-bit-equivalent BOPs budgets. Gaps are relative percentages.

Layers	$L^\star$	One-shot loss	I2LP loss	One-shot gap (%)	I2LP gap (%)
8	1.98605	1.98605	1.98605	0.00	0.00
9	2.04425	2.18178	2.04425	6.73	0.00
10	1.92966	1.99589	1.99589	3.43	3.43
11	2.09243	2.19129	2.13937	4.72	2.24
12	1.92050	2.14100	2.01861	11.48	5.11
13	1.91802	2.16958	1.91802	13.12	0.00
14	2.07676	2.32234	2.07676	11.83	0.00
15	1.99933	2.22414	1.99933	11.24	0.00
16	2.18641	2.35739	2.18641	7.82	0.00
			Mean gap	7.82	1.20

layer to take a weight/activation bit-width in $\{3, 5\}$, while keeping the remaining layers at 4-bit. The BOPs budget corresponds to uniform 4-bit quantization.

Exhaustive search, one-shot ILP, and iterative I2LP use the same quantization setup and calibration task loss, evaluated on 32 calibration images without scale updates. Both search methods start from the same uniform 4-bit reference. The one-shot ILP result is the initial unrestricted surrogate-optimal assignment, which I2LP subsequently refines using multiple update-size limits for up to 10 iterations, with early stopping.

We define the relative optimality gap as $100(L - L^\star)/L^\star$, where $L^\star$ is the minimum calibration task loss over all enumerated feasible configurations.

As shown in Table 10, iterative I2LP reduces the mean relative optimality gap from 7.82% to 1.20%, while the mean Hamming distance to an exact optimum decreases from 2.00 to 0.67. I2LP reaches an exact optimum in six of the nine subproblems, compared with one for one-shot ILP. In particular, I2LP reaches the exact optimum in all four subproblems with 13–16 selectable layers.

J Additional Mixed-Precision Baseline Comparisons

LAMPQ protocol. Table 2 compares I2LP with LAMPQ under AdaLog-based quantization and parameter-weighted average-bit budgets, unlike the RepQViT+-based BOPs comparison in Table 1(c). Following LAMPQ's bit-allocation granularity, we assign bits to qkv, proj, fc1, and fc2 in each of the 12 DeiT blocks, with the average-bit budget applied to these 48 modules. In our implementation, patch embedding and the classifier are fixed at W4/A8 and W4/A4, respectively, outside this budget.

Following the calibration settings in LAMPQ's released implementation, we use 32 images for AdaLog raw calibration. For I2LP's task-loss evaluation during bit selection, we use 1,024 ImageNet-1K images (batch size 32), matching the optimization-set size specified in LAMPQ's released implementation.

The uniform AdaLog baseline and I2LP results are from our implementation, whereas the LAMPQ results are taken from Table 3 of [17]. I2LP achieves higher Top-1 accuracy than LAMPQ by 0.47 percentage points on average at 4-bit and 6.81 points at 3-bit across DeiT-T/S/B.

QAT comparisons with InfoQ and MSQ. We also compare I2LP with recent QAT results from InfoQ [1] and MSQ [11]. Table 11 reports Top-1 accuracy and the number of QAT epochs for each method. I2LP uses 100 epochs of QAT in both settings. Although the resource budgets are matched, the bit-selection and training procedures differ. The MSQ comparison uses FP32 activations.

K Weight-Only Quantization Details on Qwen3-8B

Basic quantizer and evaluation. We evaluate Qwen3-8B [39] with bit selection on 216 linear layers in its 36 transformer blocks, comprising q/k/v/o/gate/up projections (5,133,828,096 selected weight parameters). Each selected linear layer chooses one bit-width from $\{2, 3, 4, 5, 6, 7, 8\}$. The

Table 11: Comparison with published QAT results under matched resource budgets. Top-1 accuracy (%). Gains are relative to the corresponding baseline. †Based on InfoQ's released training log.

Model	Resource budget	Method	QAT epochs	Top-1 (%)	Gain (pp)
ResNet-18	23.04 GBOPs	InfoQ	340†	69.99	–
		I2LP (ours)	100	**70.53**	+0.54
ResNet-50	10.89× W comp.	MSQ	100	75.32	–
		I2LP (ours)	100	**76.36**	+1.04

basic quantizer uses symmetric uniform weight quantization with per-output-channel scaling, where the scale is defined as $s_c = \max_j |W_{c,j}|/2^{b-1}$. The quantized integer values are rounded and clipped to $[-2^{b-1}, 2^{b-1} - 1]$. Down projections, embedding/output modules, and normalization layers remain in BF16, as do activations. The constraint applies to the parameter-weighted bit count of selected weights only; constant-precision parameters, scale metadata, packing overhead, and final whole-model memory are not included in this budget.

Selection uses eight WikiText calibration batches of 128 tokens (1,016 next-token prediction targets) and token-wise KL divergence to the BF16 teacher at temperature 1.0.

We initialize all 216 selected layers at 3-bit and use $U^{(t)} = \{1, 2, 4, 8, 16, 216\}$ from the first iteration, with a maximum of 10 iterations. We set a 60-second time limit for each ILP solve. Perplexity is evaluated on the WikiText-2 raw test split. The six-task score is the unweighted mean of zero-shot `acc_norm` for PIQA, ARC-Easy, ARC-Challenge, and HellaSwag, and `acc` for WinoGrande and BoolQ.

Table 12: Qwen3-8B W3-budget results with and without RMSNorm-only distillation fine-tuning. WikiText-2 perplexity and zero-shot six-task accuracy are reported.

Fine tuning	Bit allocation	WikiText-2 PPL ↓	Six-task avg. (%) ↑
None	Uniform W3	41.52	51.77
None	I2LP W3-budget	**18.12**	**57.64**
RMSNorm only	Uniform W3	12.99	66.14
RMSNorm only	I2LP W3-budget	**12.41**	**67.68**

RMSNorm-only fine-tuning updates only the normalization parameters using BF16-teacher distillation, without a ground-truth cross-entropy term. Both uniform W3 and I2LP models use the same fine-tuning settings: the first 5% of the WikiText-2 raw training split, 2,048-token blocks, batch size 1, and one epoch (62 optimization steps). We use AdamW with a maximum learning rate of 5×10^{-3}, zero weight decay, and a cosine learning-rate schedule with 25% warmup. I2LP retains an advantage over uniform W3 after fine-tuning, although the performance gap becomes smaller.

Extension to AWQ-style quantization. We also test I2LP with AWQ-style scaling and reparameterization [20], combined with asymmetric per-output-channel weight quantization. In this additional experiment, all seven projections (q/k/v/o/gate/up/down) are targeted. The q/k/v, gate/up/down, and o projections form three distinct bit-allocation groups per transformer block, yielding 108 variables across 36 blocks. Candidate bits range from 2 to 6, with a parameter-weighted 3-bit budget over the seven selected projection types. We use C4 for selection and final AWQ-style calibration, without fine-tuning. Activations and other modules remain in BF16.

Table 13: AWQ-style weight-only W3 evaluation on Qwen3-8B.

Configuration	WikiText-2 PPL ↓
Uniform AWQ-style W3	15.0147
I2LP, AWQ-style W3 budget	**13.9497**

I2LP reduces WikiText-2 perplexity by 7.09% relative to uniform AWQ-style W3, while the zero-shot six-task averages remain similar (62.40% for I2LP vs. 62.48% for uniform W3). These seven-

projection AWQ-style results use a different quantization setup from the six-projection basic-quantizer experiments.

L Measured-Latency Mixed-Precision Evaluation

We evaluate ResNet-18 on an NVIDIA RTX PRO 6000 Blackwell GPU using TensorRT 10.8. Bit selection is applied to 21 layers: the stem convolution, the convolutions in the residual blocks (including downsampling), and the final fully connected layer. Other operations are retained from the FP32 ONNX graph, although TensorRT may fuse them with adjacent layers.

The candidate precisions are FP16, INT8, and FP8 for convolutional layers, and FP16 and INT8 for the final fully connected layer. The all-target INT8 and FP8 baselines assign the corresponding precision to all 21 layers, including the final fully connected layer. I2LP selects 12 FP16 and 9 FP8 layers, with the final fully connected layer assigned FP16. Although our ONNX/TensorRT pipeline does not support mixed INT8/FP8 assignments, the selected FP16/FP8 configuration can be compiled and measured.

Latency cost. We construct the latency cost from TensorRT measurements. Let t_0 denote the measured latency of the FP32 model and $t_i^{(p)}$ the latency when only layer i is changed to precision p. The single-layer latency difference is

$$d_i^{(p)} = t_i^{(p)} - t_0. \tag{16}$$

To account for interactions between layers, we also measure the latency $t_{ij}^{(p)}$ when layers i and j both use precision p, with all other target layers remaining FP32. The interaction cost is defined as

$$c_{ij}^{(p)} = \left(t_{ij}^{(p)} - t_0\right) - d_i^{(p)} - d_j^{(p)}. \tag{17}$$

We focus on two types of local interactions: the two convolutions within a residual block (pair) and convolutions across adjacent residual blocks (bridge). These interactions are particularly relevant because quantization boundaries and TensorRT fusion can depend on the precisions of neighboring layers. Pair corrections are included for FP16, INT8, and FP8, while bridge corrections are included only for FP8, whose measured cross-block interactions are substantial.

The resulting latency surrogate is

$$\begin{aligned}\widehat{t}(\mathbf{x}) = t_0 &+ \sum_i \sum_{p\in\mathcal{P}_i} d_i^{(p)} x_i^{(p)} \\ &+ \sum_{(i,j)\in\mathcal{E}_{\text{pair}}} \sum_{p\in\{\text{FP16,INT8,FP8}\}} c_{ij}^{(p)} z_{ij}^{(p)} \\ &+ \sum_{(i,j)\in\mathcal{E}_{\text{bridge}}} c_{ij}^{(\text{FP8})} z_{ij}^{(\text{FP8})},\end{aligned} \tag{18}$$

where $\mathcal{P}_i$ is the set of candidate precisions for layer i, $x_i^{(p)}$ indicates its selected precision, and $z_{ij}^{(p)} = x_i^{(p)} x_j^{(p)}$. The product terms are linearized using auxiliary binary variables. By restricting the corrections to neighboring layers, the eight pair edges and seven bridge edges introduce only 31 additional binary variables. We set the latency constraint to -0.373317 ms relative to FP32, based on the measured latency difference between the FP32 and all-target INT8 models.

Evaluation settings. I2LP starts from an all-target INT8 assignment. The first iteration uses $M = 21$, while subsequent iterations consider $M \in \{2, 4, 8, 16, 21\}$. The task loss is evaluated using 1,024 ImageNet images.

For accuracy evaluation, INT8 and FP8 use fake quantization with symmetric per-tensor activation quantization and symmetric per-output-channel weight quantization. FP16 layers use FP16 casts. No QAT or reconstruction is applied.

For TensorRT, FP8 uses the E4M3FN format, with activation scales calibrated using 128 ImageNet validation images. The PyTorch and TensorRT quantization scales are determined independently. We

evaluate ImageNet Top-1 accuracy in PyTorch and measure end-to-end latency and throughput using TensorRT's `trtexec`. The reported accuracies are therefore not measurements from the compiled TensorRT engines. Latency is reported as the mean GPU Compute Time, using a 3-second warmup, a 30-second measurement duration, and `--useSpinWait`.

Table 14: ResNet-18 results on RTX PRO 6000 Blackwell. Latency and throughput are measured using TensorRT, while ImageNet Top-1 accuracy is evaluated in PyTorch. All-target baselines assign the specified precision to all 21 target layers.

Precision	Latency (ms) ↓	Images/s ↑	Top-1 (%) ↑
FP32 baseline	0.5618	1775.94	72.54
All-target INT8	0.1884	5275.30	65.88
All-target FP8	0.1448	6848.82	69.91
I2LP FP16/FP8	0.1560	6358.29	72.33

Compared with all-target FP8, I2LP improves Top-1 accuracy by 2.42 percentage points at 7.7% higher measured latency. Compared with all-target INT8, I2LP improves Top-1 accuracy by 6.45 percentage points while reducing latency by 17.2%. I2LP achieves a $3.60\times$ measured speedup over FP32 with only a 0.21 percentage-point drop in Top-1 accuracy.

The augmented latency surrogate predicts 0.1839 ms for the selected assignment, whereas the compiled engine measures 0.1560 ms. This difference suggests that the single-layer, pair, and bridge measurements do not fully capture full-network interactions and TensorRT optimizations.

M Scope, Limitations, and Broader Impacts

Scope and limitations. This work primarily targets mixed-precision bit selection for vision models, with an additional evaluation on weight-only quantization of a large language model. Most experiments use model size and BOPs as resource constraints, while Appendix L evaluates a measured-latency constraint on a specific GPU. Hardware performance remains dependent on the target device, compiler, and implementation. I2LP also requires calibration data and additional task-loss evaluations.

Broader impacts. Efficient mixed-precision quantization can reduce deployment cost and improve accessibility of vision models on resource-constrained devices, but may also make arbitrary downstream vision models easier to deploy. The societal impact therefore depends on the underlying model, dataset, and application.

N Existing Assets and Licenses

We use existing datasets, pretrained checkpoints, and software libraries for research evaluation. Table 15 summarizes the main external assets used in our experiments and their corresponding licenses or terms of use. Methods that are only cited as baselines, or algorithms that are implemented by ourselves from the corresponding papers, are not listed as external code assets. In particular, LSQ and QDrop are implemented by ourselves based on the corresponding papers and are cited accordingly.

Table 15: Existing external assets used in this work.

Asset	Use in this paper	License / terms
ImageNet	Image classification evaluation	Official ImageNet access terms
COCO 2017	Object detection evaluation	COCO terms of use
DINOv3 checkpoints	Pretrained vision backbone	Official DINOv3 release terms
DETR source code and checkpoint	Object detection quantization	Apache License 2.0
PyTorch	Implementation framework	BSD-style license
torchvision / timm	Model definitions and checkpoints	Official open-source licenses

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state that the paper addresses surrogate–task mismatch in mixed-precision bit-width selection and proposes I2LP with update-size control. The claimed scope is consistent with the empirical evaluation on PTQ and QAT settings across CNNs, ViTs, and DETR.

2. **Limitations**

   Answer: [Yes].

   Justification: Appendix M discusses the scope and limitations of the work, including the additional task-loss evaluations required by I2LP, the primary use of model size and BOPs as resource constraints, and the measured-latency evaluation on a specific GPU.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

   Justification: The theorem and corollaries are stated with their assumptions, and their proofs are provided in Appendix B. The theoretical results are used to motivate the relation between additive surrogates, local updates, and surrogate–task mismatch.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: The paper specifies the proposed algorithms, datasets, model architectures, calibration sample sizes, candidate bit-width sets, initialization settings, task loss, and evaluation protocols. These details provide a reasonable path to reproducing the main experimental results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No].

   Justification: The paper states that code will be released, but the current submission does not provide anonymized code or full reproduction scripts. The experiments use standard public datasets, ImageNet and COCO 2017.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Section 4 and Appendix C specify the datasets, model architectures, calibration sample sizes, bit-width candidates, initial high-bit references, task-loss definition, update-size sets, evaluation metrics, and QAT protocol. For QAT, we follow the experimental protocol of MetaMix after bit selection, including the teacher models used for distillation.

7. **Experiments compute resources**

   Answer: [Yes].

Justification: Appendix H reports the compute environment, including GPU type and memory, as well as representative wall-clock time, GPU-hours, and ILP solver statistics. The reported results show that solver time is negligible compared with surrogate construction and candidate evaluation.

8. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The research uses standard public benchmark datasets and does not involve human subjects, private data, surveillance data, or high-risk model release. We have reviewed the NeurIPS Code of Ethics and believe the work conforms to it.

9. **Broader impacts**

   Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

   Answer: [Yes].

   Justification: Appendix M discusses positive impacts, such as reducing deployment cost and improving accessibility of vision models on resource-constrained devices, as well as potential negative impacts from making arbitrary downstream vision models easier to deploy.

10. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release high-risk pretrained models, image generators, scraped datasets, or other assets that require special safeguards for misuse prevention.

11. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: Appendix N lists the main external datasets, pretrained checkpoints, source code, and software libraries used in the experiments, including ImageNet, COCO 2017, DINOv3 checkpoints, the official DETR source code and checkpoint, PyTorch, torchvision, and timm, together with their licenses or terms of use. Algorithms implemented by ourselves from cited papers, such as LSQ, QDrop and RepQ-ViT, are cited but not listed as external code assets.

12. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: The paper introduces a new algorithm but does not currently release a new dataset, pretrained model, or benchmark asset. Source code is planned for release and is addressed under the open access to data and code item.

13. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects.

14. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval or equivalent review is not applicable.

15. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core methodology does not use LLMs as an important, original, or non-standard component. Any use of LLMs for writing, editing, or formatting does not affect the scientific methodology, experimental results, or originality of the work.