REVISION: REPROGRAMMING IMAGE GENERATORS FOR DENSE VISUAL PERCEPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent image generators exhibit rich visual priors that can support dense visual perception. However, existing approaches typically encode multiple perception tasks into a shared parameterization, making joint training sensitive to the mixture and weighting of heterogeneous data and prone to cross-task interference. To address this issue, we introduce ReVision, an instruction-conditioned parameter generation framework built on a frozen multimodal understanding model and a frozen image generator. Given an image and a free-form instruction, the understanding model jointly encodes them into a multimodal image–instruction representation, which a trainable parameter generator maps to transient low-rank updates for the generator. Each image–instruction pair induces a distinct set of transient prediction parameters, enabling instance-specific adaptation while both pretrained backbones remain unchanged. ReVision requires no task-specific heads, explicit task identifiers, or stored adapters. A shared RGB output interface unifies segmentation, depth estimation, and surface-normal prediction. Across diverse dense prediction benchmarks, ReVision achieves competitive performance with state-of-the-art methods while supporting flexible task specification through natural-language instructions. Our analysis further reveals substantial gradient disagreement across tasks, highlighting the conflict introduced when heterogeneous objectives share a single parameterization. Meanwhile, the generated updates exhibit structured task- and input-dependent organization in weight space, further validating instance-level parameter generation for multi-task perception.

1 INTRODUCTION

Visual perception requires models to recognize objects, understand scene semantics, and reason about spatial structure and geometry. Visual representation learning has traditionally relied on discriminative supervision (Krizhevsky et al., 2012; Dosovitskiy et al., 2020; Dehghani et al., 2023), contrastive learning (Chen et al., 2020; He et al., 2020; Radford et al., 2021), self-distillation (Zhou et al., 2022; Caron et al., 2021), or reconstruction-based pretraining (Bao et al., 2021; He et al., 2022). In parallel, recent advances in image generation (Labs et al., 2025; Liu et al., 2026a; Gao et al., 2025; Zhao et al., 2026a) have substantially expanded what generative models can represent. SOTA generators synthesize complex scenes with high visual fidelity, semantic alignment, and fine-grained control, suggesting that their representations capture objects, geometry, semantics, and their interactions. These works have also demonstrated potential for visual perception.

Recent work has explored this possibility by repurposing pretrained image generators for visual perception. Early studies extracted intermediate diffusion features for segmentation, correspondence, and related recognition tasks (Zhao et al., 2023; Xu et al., 2023; Tang et al., 2023). Later methods formulated dense prediction directly as image generation, adapting pretrained generators to produce depth maps, surface normals, segmentation masks, and other spatially structured outputs (Ke et al., 2024; Fu et al., 2024; Zhao et al., 2026b; Xu et al., 2025b). More recent approaches extend this formulation to multiple dense prediction tasks through a shared visual output space and natural-language instructions (Gabeur et al., 2026; Zhao et al., 2026c). Related multimodal foundation models combine visual understanding and generation within a common architecture (Sun et al., 2024; Wang et al., 2024; Chameleon Team, 2024; Wu et al., 2025; Deng et al., 2025), with SenseNova-Vision further extending this paradigm to heterogeneous vision tasks using language for symbolic outputs and images for spatial predictions (Han et al., 2026). Although these works estab-

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.

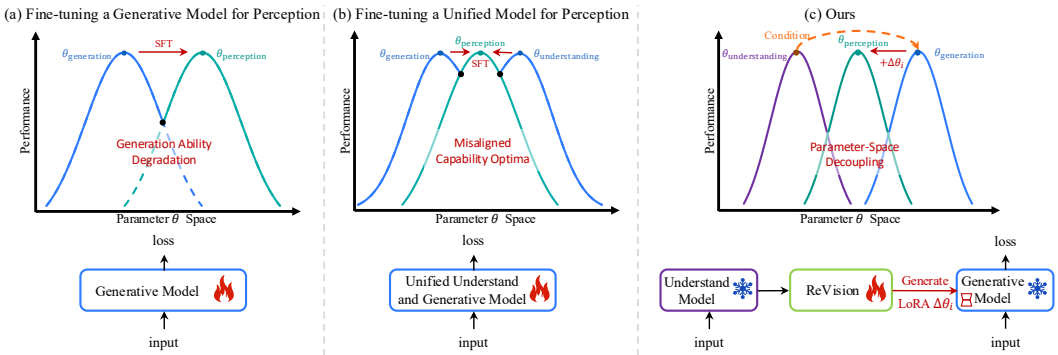


Figure 1: **Parameter-space decoupling for visual perception.** (a) Fine-tuning repurposes a pretrained generator through a shared set of optimized parameters. (b) Unified models encode understanding, generation, and perception within one parameterization. (c) ReVision keeps both backbones frozen and generates instruction-conditioned low-rank updates for each input.

lish image generation as a viable interface for visual perception, perception and generation remain coupled through shared model parameters, as illustrated in Fig. 1. Fine-tuning a pretrained generator for perception alters the parameter configuration that supports its original generative capability, often requiring careful balancing between generation and perception data. Unified understanding–generation models face a similar limitation, since understanding, generation, and perception must coexist within a single parameterization even when their preferred solutions differ. Extending such models therefore still relies on further parameter optimization. We ask: *Can a pretrained image generator be dynamically repurposed for visual perception without updating its parameters?*

We answer this question with ReVision, which connects a frozen multimodal understanding model to a frozen image generator through a trainable parameter generator. Given an image and a natural-language instruction, the understanding model produces a multimodal image–instruction representation, from which the parameter generator synthesizes transient, instance-specific low-rank updates for the generator. A separate update is generated for each image–instruction pair and used only for the current prediction, leaving both pretrained backbones unchanged. Different inputs can therefore instantiate different predictors without explicit task identifiers, task-specific heads, or stored adapters. Unlike prior hypernetwork-based methods that condition parameter generation on image features, predefined tasks, classifier descriptions, or reusable concepts (Tian et al., 2020; Nirkin et al., 2021; Zhang et al., 2021a; Liu et al., 2022; Tian et al., 2025; Li et al., 2026; Charakorn et al., 2025; Liang et al., 2025), ReVision generates parameters from multimodal image–instruction representations and applies them to a frozen image generator. We evaluate ReVision on segmentation, depth estimation, and surface-normal prediction, all expressed in a shared RGB output space, achieving competitive performance with state-of-the-art methods while supporting flexible task and output specification through natural-language instructions. We further analyze the generated updates and observe substantial cross-task gradient disagreement under shared-parameter multi-task learning, together with structured task- and input-dependent organization in weight space, supporting the effectiveness of instance-level parameter generation. Our main contributions are:

- We introduce ReVision, an instruction-conditioned parameter generation framework that maps multimodal image–instruction representations from a frozen understanding model to transient, instance-specific low-rank updates for a frozen image generator.
- ReVision enables segmentation, depth estimation, and surface-normal prediction within a shared RGB output space while achieving performance competitive with state-of-the-art methods without task-specific heads, explicit task identifiers, or stored adapters.
- Our analysis reveals cross-task gradient conflict under shared-parameter multi-task learning and structured task- and input-dependent organization of the generated updates in weight space, supporting instance-level parameter generation.

2 Related Work

Image Generative Models for Dense Perception. Pretrained image generators encode useful priors for geometry, semantics, and spatial structure. Early work exploited intermediate diffusion features

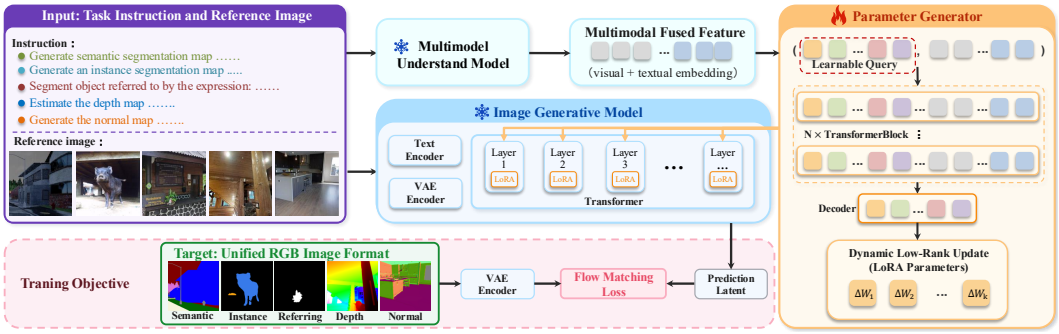


Figure 2: **Overview of ReVision.** A frozen multimodal model extracts an image–instruction representation that conditions a parameter generator to produce input-dependent low-rank updates for a frozen image generator. Diverse dense perception tasks are unified as RGB generation, with only the parameter generator trained using a flow-matching objective.

for segmentation and correspondence (Zhao et al., 2023; Xu et al., 2023; Tang et al., 2023), while subsequent methods repurposed generative models for dense prediction, especially depth and surface normals (Ke et al., 2024; Fu et al., 2024; Xu et al., 2025a; He et al., 2025b;a). Recent methods further unify heterogeneous perception tasks through image-space outputs and prompt-based task specification (Zhao et al., 2026b; Xu et al., 2025b; Khangaonkar & Pirsiavash, 2025; Zhao et al., 2026c), with stronger generators showing effective perception under limited adaptation (Gabeur et al., 2026; Liu et al., 2026b; Yang et al., 2026a). These methods still rely on a fixed set of adapted parameters. ReVision instead keeps the generator frozen and generates an instance-specific parameter update from each image–instruction pair.

Unified Models for Understanding and Generation. Prior work has unified visual understanding and generation within a single model. Early approaches cast heterogeneous vision tasks into common sequence or image prediction interfaces (Chen et al., 2022; Lu et al., 2023; 2024; Mizrahi et al., 2023; Bachmann et al., 2024; Wang et al., 2023; Geng et al., 2024). Recent multimodal models integrate understanding and generation in shared architectures (Sun et al., 2024; Wang et al., 2024; Chameleon Team, 2024), using shared representations, mixed objectives, or specialized experts to handle their different requirements (Wu et al., 2025; Zhou et al., 2025; Xie et al., 2025; Deng et al., 2025). They can also be adapted to heterogeneous perception tasks through instruction tuning (Han et al., 2026). ReVision keeps both models frozen and uses a parameter generator to translate multimodal representations from the understanding model into instance-specific low-rank updates for the image generator, coupling them through generated parameters rather than shared weights.

Conditional Parameter Generation. Hypernetworks generate target-network parameters from conditioning signals, offering an alternative to fixed parameterization (Ha et al., 2017; Jia et al., 2016). They have been used for dynamic prediction (Tian et al., 2020; Wang et al., 2020b; Zhang et al., 2021b), task-conditioned adaptation (Zhang et al., 2021a; Liu et al., 2022), and lightweight adapter generation (Hu et al., 2022; Karimi Mahabadi et al., 2021; Phang et al., 2023). Recent methods predict LoRA weights from language or visual conditions for transfer, classification, and generative personalization (Charakorn et al., 2025; Duan & Chen, 2026; Tian et al., 2025; Li et al., 2026; Ruiz et al., 2024; Team, 2026). ReVision instead generates low-rank updates from free-form image–instruction representations and applies them to a frozen image generator for dense prediction. These updates define an instance-specific perception function rather than a fixed task adapter, classifier head, or personalization module.

3 Method

3.1 Motivation and Formulation

Let G_θ denote a pretrained image generator with parameters θ, and let a dense-prediction sample be (I, P, y), where I is the input image, P is a free-form natural-language instruction, and y is the target output. Existing approaches typically learn a shared parameter set:

$$\theta^\star = \arg\min_\theta \mathbb{E}_{(I,P,y)\sim\mathcal{D}} \left[\mathcal{L}\big(G_\theta(I,P), y\big)\right]. \tag{1}$$

Although the desired behavior may be specified through an instruction or task token, all inputs are ultimately processed by the same optimized parameters $\theta^\star$. Heterogeneous generation and perception behaviors must therefore coexist within a single parameter configuration, which can introduce interference across tasks. We instead learn a parameter generator H_ψ that maps an input-dependent multimodal representation $\mathbf{z}$ to parameter updates for the generator:

$$\theta(\mathbf{z}) = \theta \oplus H_\psi(\mathbf{z}), \qquad \hat{y} = G_{\theta(\mathbf{z})}(I, P), \tag{2}$$

where $\oplus$ denotes applying the generated update to the pretrained parameters. The parameters θ remain frozen, while each conditioning representation $\mathbf{z}$ induces a transient parameterization. Different conditions can thus instantiate different predictors without modifying the pretrained generator or storing task-specific parameters. Generating the full parameter set of an image generator is prohibitively expensive, so we generate only LoRA updates (Hu et al., 2022) for a selected set of layers.

3.2 Parameter Generator

Conditioning input. The parameter generator requires a conditioning representation that captures both the visual content of the input and the task specified by the instruction. Although pretrained image generators encode rich visual structure, their representations are optimized primarily for image synthesis. We therefore use a separate frozen multimodal understanding model, `Qwen3.5-9B-Base` (Qwen Team, 2026), to provide task-aware representations for perception. These representations condition the parameter generator, which steers the pretrained visual representations of the generator toward the requested dense prediction behavior. Specifically, the input image and instruction are jointly processed by the multimodal understanding model. We extract hidden states from an intermediate layer, where visual and textual information has been fused while remaining less specialized toward next-token prediction. The conditioning feature $\mathbf{Z}$ is defined as

$$\mathbf{Z} = W_{\text{proj}} \Phi_k(I, P), \tag{3}$$

where $\Phi_k(I, P)$ denotes the multimodal hidden states from layer k, and W_{proj} maps them to the hidden dimension of the parameter generator. We set $k = 16$ in all experiments.

Query-based parameter generation. To generate LoRA parameters for the image generator, we introduce a bank of learnable parameter queries. Let S denote the number of LoRA injection sites in the frozen image generator, r the LoRA rank, and d the hidden dimension of the parameter generator. We define $Q \in \mathbb{R}^{2Sr \times d}$, where each injection site is assigned r queries for LoRA-A and another r queries for LoRA-B. The queries are learned jointly with the parameter generator and serve as site- and factor-specific parameter slots. They are then concatenated with the conditioning feature $\mathbf{Z}$ and processed by N Transformer blocks $\mathcal{T}$:

$$[\mathbf{Q}'; \mathbf{Z}'] = \mathcal{T}([\mathbf{Q}; \mathbf{Z}])\,. \tag{4}$$

Each block consists of parallel multi-head self-attention and gated MLP branches, allowing the parameter queries to interact extensively with the conditioning tokens throughout the Transformer stack. After the final block, we retain only the updated query features $Q' \in \mathbb{R}^{2Sr \times d}$ and discard the transformed conditioning features Z'. We then partition Q' according to the injection sites and LoRA factors. For each injection site s, the corresponding query features are denoted by $X_s^A, X_s^B \in \mathbb{R}^{r \times d}$, which are used to generate the LoRA-A and LoRA-B parameters, respectively. The two groups are decoded independently by site-specific linear projections,

$$A_s = f_s^A(X_s^A), \qquad B_s = f_s^B(X_s^B), \qquad s = 1, \ldots, S, \tag{5}$$

where f_s^A and f_s^B map the query features to the dimensions required by the LoRA-A and LoRA-B matrices at injection site s. This results in $2S$ decoding heads, each producing one LoRA factor for its site. The generated LoRA parameters are then injected into the frozen image generator.

LoRA injection sites. We use `FLUX.2-klein-base-9B` (Labs, 2025) as the frozen image generator, consisting of 8 double-stream and 24 single-stream blocks. Generated LoRA parameters are injected into selected projection layers throughout the backbone, with eight sites per double-stream block and two per single-stream block, yielding $S = 112$ injection sites in total. For each site, the parameter generator produces the corresponding LoRA-A and LoRA-B matrices. We use rank $r = 16$ and set $\alpha/r = 1$ for all sites, while the base generator remains frozen. Detailed model and injection settings are provided in the Appendix C.

3.3 Unified RGB Output Interface

To keep the frozen generator within its native output space, we follow (Gabeur et al., 2026) and represent all dense-prediction targets as RGB images. The generator therefore always predicts an RGB image, while the instruction specifies the desired perception task and output convention.

Segmentation. We encode semantic, instance, and referring segmentation as color masks. For semantic segmentation, each class present in an image is assigned a distinct color, with the class-to-color mapping explicitly specified in the instruction. Both the color assignment and its textual representation, including color names, RGB values, and hexadecimal codes, are randomized across samples to avoid a fixed dataset-level mapping. For instance segmentation, colors identify individual object instances and a separate palette is constructed for each image. For referring segmentation, the referred region and background are encoded with two contrasting colors specified by the instruction. We retain only expressions referring to a single instance to ensure an unambiguous target.

Depth estimation. We encode relative depth as an RGB image using a continuous color map $\Gamma : [0,1] \to [0,1]^3$ defined along a Hamiltonian path through the eight vertices of the RGB cube. For each image, let ℓ and h denote the 2nd and 98th percentiles over valid pixels. We compute

$$u = \mathrm{clip}\left(\frac{x-\ell}{h-\ell}, 0, 1\right), \quad \rho = \tau + (1-\tau)u, \quad \Gamma(\rho) = v_j + (7\rho - j)(v_{j+1} - v_j), \quad j = \lfloor 7\rho \rfloor, \tag{6}$$

where x denotes depth or inverse depth, oriented such that u increases with distance, and $\tau > 0$ reserves black for invalid pixels. The predicted RGB value is decoded by projection onto the same path. This per-image normalization removes absolute scale and shift, allowing metric and scale-ambiguous sources to be trained jointly. At evaluation time, predictions are aligned to the ground truth by least-squares scale and shift. We discard samples with negligible relative depth variation, $(h-\ell)/\max(|h|,|\ell|) < \epsilon$, to avoid amplifying sensor or reconstruction noise.

Surface normal estimation. Surface normals are encoded directly as RGB images by linearly mapping each component from $[-1,1]$ to $[0,255]$. Predictions are mapped back to $[-1,1]$ and normalized to unit length, while invalid pixels are encoded as black. Normals from all sources are converted to a common camera coordinate system, with X_{cam} pointing right, Y_{cam} down, and Z_{cam} into the scene. Since surface normals are scale invariant, both metric and scale-ambiguous geometry sources can be used jointly for training. Data generation details are provided in the Appendix B.

3.4 Training

Objective. Given noise $\varepsilon \sim \mathcal{N}(0, I)$ and timestep $t \in (0,1)$, let $\mathbf{e}^{\mathrm{gt}}$ denote the VAE-encoded ground-truth output, $\mathbf{e}_{\mathrm{ref}}$ the VAE-encoded reference image I, and $\mathbf{e}_p$ the text embedding of the instruction P. We further denote by x_t the denoising target at timestep t. With the image generator frozen, we only optimize the parameter generator H_ψ:

$$\mathcal{L}(\psi) = \left\| G_{\theta \oplus H_\psi(\mathbf{z})}\left(\mathbf{x}_t, \mathbf{e}_{\mathrm{ref}}, \mathbf{e}_p, t\right) - \left(\varepsilon - \mathbf{e}^{gt}\right) \right\|_2^2. \tag{7}$$

We sample t from a logit-normal distribution (Esser et al., 2024) and apply a fixed resolution-independent shift of $e^{0.8} \approx 2.2255$.

Training schedule. We train progressively by resolution, using 256^2 for 10k steps on four tasks, followed by 512^2 for 20k steps with referring segmentation added. We transfer the raw hypernetwork weights between stages and restart the optimizer and learning-rate schedule. Both stages use AdamW with a peak learning rate of 2×10^{-5}, uniform task sampling, and mixed-task batches. We maintain an EMA with decay 0.999 and use the EMA checkpoint for evaluation. Full hyperparameters are provided in Appendix A.

Training data. All tasks are drawn from public corpora and tiled into fixed-resolution crops. Semantic segmentation data are collected from ADE20K (Zhou et al., 2017), COCO-Stuff-164K (Caesar et al., 2018), Mapillary Vistas v2 (Neuhold et al., 2017), and BDD7K (Yu et al., 2020), while instance segmentation uses LVIS (Gupta et al., 2019), Open Images V7 (Kuznetsova et al., 2020), and GranD (Rasheed et al., 2024). We relabel these datasets with SAM 3 (Carion et al., 2026), followed by deduplication and quality filtering, to obtain fine-grained masks. Referring segmentation uses the

original instructions from GranD (Rasheed et al., 2024) together with SAM 3 relabeling. Depth estimation and surface-normal estimation are treated as separate tasks, both using Omnidata (Eftekhar et al., 2021) and TartanAir (Wang et al., 2020a). No evaluation dataset is included in training.

4 Experiments

4.1 Comparison Results

For a controlled comparison, we additionally train two variants using the same image generator. *Task-specific LoRA* trains one adapter per task, while *Multi-task LoRA* jointly trains a single adapter on all tasks. ReVision uses the same multi-task data but generates task- and input-dependent updates from the instruction. Qualitative results are provided in Appendix F.

Semantic Segmentation. We evaluate zero-shot semantic segmentation on Cityscapes val., whose images, 19-class `trainId` taxonomy, and label conventions are absent from training. At inference time, both the target categories and their color mappings are specified through natural-language instructions, allowing open-vocabulary task specification without a fixed label set. As shown in Tab. 1, ReVision reaches 72.75 mIoU and outperforms both LoRA baselines. The lower performance of Multi-task LoRA relative to Task-specific LoRA also suggests interference when heterogeneous tasks share a single fixed update.

Table 1: Semantic segmentation on Cityscapes.

Method	Zero-shot	mIoU ↑
SegMan-L (Fu et al., 2025b)	✗	84.2
SenseNova-Vision (Han et al., 2026)	✗	71.2
APE-D (Shen et al., 2024)	✓	44.2
OpenSeeD (Zhang et al., 2023)	✓	47.8
X-Decoder (Zou et al., 2023)	✓	52.0
SAM 3 (Carion et al., 2026)	✓	65.2
Vision-Banana (Gabeur et al., 2026)	✓	69.9
Task-specific LoRA	✓	<u>70.07</u>
Multi-task LoRA	✓	64.68
ReVision (Ours)	✓	**72.75**

Instance Segmentation. We further evaluate on the open-vocabulary SA-Co/Gold benchmark, where the model must separate an unknown number of instances of the queried concept. As shown in Tab. 2, ReVision achieves 40.17 cgF_1, 0.74 IL_MCC, and 54.3 pmF_1, outperforming both LoRA baselines and several representative zero-shot methods. The gains are largest on cgF_1 and pmF_1, indicating improved instance separation and mask quality. IL_MCC is similar across our three variants, suggesting that instance-existence prediction is mainly determined by the shared `Qwen3.5-9B` component. A gap remains to Vision Banana, particularly on cgF_1, likely reflecting differences in the underlying foundation model.

Table 2: Instance segmentation on SA-Co/Gold.

Model	Zero-shot	cgF_1 ↑	IL_MCC ↑	pmF_1 ↑
SAM 3 (Carion et al., 2026)	✗	54.1	0.82	66.1
SAM 3 (Carion et al., 2026) + Llama 3.2 (ft)	✗	61.2	0.86	70.8
gDino-T (Liu et al., 2024)	✓	3.3	0.15	16.2
LLMDet-L (Fu et al., 2025a)	✓	6.5	0.21	27.3
Gemini 2.5 (Comanici et al., 2025)	✓	13.0	0.29	46.1
APE-D (Shen et al., 2024)	✓	16.4	0.40	36.9
DINO-X (Ren et al., 2024)	✓	21.3	0.38	<u>55.2</u>
OWLv2 (Minderer et al., 2023)	✓	24.6	0.57	42.0
Vision Banana (Gabeur et al., 2026) + Gemini 3.1 Flash-Lite	✓	**47.5**	**0.84**	**56.0**
Task-specific LoRA + Qwen3.5-9B	✓	33.95	<u>0.74</u>	45.7
Multi-task LoRA + Qwen3.5-9B	✓	30.71	<u>0.74</u>	41.4
ReVision (Ours) + Qwen3.5-9B	✓	<u>40.17</u>	<u>0.74</u>	54.3

Referring Segmentation. Referring segmentation identifies a target from a free-form description, requiring finer grounding and relational reasoning than fixed-category segmentation. As shown in Tab. 3, ReVision reaches 53.8 cIoU, outperforming both LoRA baselines and approaching the RefCOCOg-supervised LoRA baseline. This result highlights the ability of instance-conditioned parameter generation to translate open-ended language descriptions into pixel-level predictions. A gap remains to Vision Banana, which may stem from the stronger base generator. In particular, GenMask (Yang et al., 2026b) shows that mask generation benefits from segmentation-specific noise and timestep sampling, whereas ReVision retains the original generative schedule.

Table 3: Referring expression segmentation results on RefCOCOg val (UMD).

Model	Zero-shot	cIoU ↑
HyperSeg-Phi2-2.7B (Wei et al., 2024)	✗	79.4
X-SAM-Phi3-3.8B (Wang et al., 2026)	✗	83.8
SenseNova-Vision (Han et al., 2026)	✗	80.3
HybridGL (Liu & Li, 2025)	✓	51.3
LocalizationHeads-LLaVA-1.5-13B (Kang et al., 2025)	✓	67.7
SAM 3 (Carion et al., 2026) + Gemini 2.5 Pro	✓	<u>73.4</u>
Vision Banana (Gabeur et al., 2026)	✓	**73.8**
RefCOCOg-supervised LoRA	✗	54.7
Task-specific LoRA	✓	53.1
Multi-task LoRA	✓	46.8
ReVision (Ours)	✓	53.8

Depth Estimation. Generated depth images are decoded and evaluated with affine-invariant metrics. As shown in Tab. 4, ReVision generalizes well across indoor and outdoor benchmarks, with strong

Table 4: **Comparison on monocular depth estimation.** *SenseNova-Vision is not strictly zero-shot due to dense geometry supervision and possible evaluation-source overlap. ‡Vision Banana reports metric-depth without per-image affine alignment and is included for reference.

Method	Zero-shot	AbsRel ↓ / δ_1 ↑					
		NYUv2	KITTI	ETH3D	ScanNet	DIODE	iBims1
Marigold (Ke et al., 2024)	✓	5.5 / 96.4	9.9 / 91.6	6.5 / 95.9	6.4 / 95.2	30.8 / 77.3	- / -
DICEPTION (Zhao et al., 2026b)	✓	6.1 / 96.0	6.9 / 94.9	5.0 / 97.5	7.2 / 94.4	28.9 / 72.2	- / -
FE2E (Wang et al., 2025)	✓	4.1 / **97.7**	**6.6** / **96.0**	3.8 / **98.7**	4.4 / 97.5	22.8 / **81.2**	- / -
Lotus-2 (He et al., 2025a)	✓	4.1 / 97.6	6.7 / 94.5	4.6 / 98.1	4.2 / 97.6	**22.1** / 75.2	- / -
Vision Banana‡ (Gabeur et al., 2026)	✓	8.1 / 94.8	10.7 / 91.5	10.3 / 93.5	- / -	- / -	7.8 / 93.4
SenseNova-Vision (Han et al., 2026)	✗*	4.0 / 98.1	5.9 / 95.9	4.3 / 97.4	3.9 / 98.0	20.6 / 76.4	- / -
Task-specific LoRA	✓	5.7 / 95.5	22.7 / 70.8	5.2 / 96.9	5.1 / 96.8	36.3 / 72.6	4.9 / 96.4
Multi-task LoRA	✓	5.9 / 95.3	19.4 / 75.4	6.5 / 95.3	5.0 / 97.0	35.3 / 73.2	5.5 / 95.8
ReVision	✓	**4.0** / 97.5	9.1 / 93.2	**3.7** / 98.1	**3.9** / **98.0**	24.5 / 79.6	**3.9** / **98.0**

Table 5: **Comparison on surface normal estimation.** We report mean and median angular errors (↓). *Lotus-2-Normal is trained on Virtual KITTI 2, so its VKitti result is not zero-shot. ‡SenseNova-Vision includes ScanNetV2 in training and is not strictly benchmark-disjoint.

Methods	Zero-shot	Indoor								Outdoor	
		Average		NYUv2		DIODE-indoor		ScanNet		VKitti	
		mean ↓	median ↓	mean ↓	median ↓	mean ↓	median ↓	mean ↓	median ↓	mean ↓	median ↓
Marigold (Ke et al., 2024)	✓	19.606	11.828	20.864	11.134	16.671	12.084	21.284	12.268	–	–
DSINE (Bae & Davison, 2024)	✓	17.017	10.190	16.4	8.4	18.453	13.871	16.2	8.3	28.9	9.9
StableNormal (Ye et al., 2024)	✓	17.168	10.028	19.707	10.527	**13.701**	**9.46**	18.098	10.097	–	–
Lotus-2-Normal (He et al., 2025a)	✗*	16.558	–	16.9	N/A	18.575	N/A	14.2	N/A	28.894	9.677
SenseNova-Vision (Han et al., 2026)	✗‡	14.167	–	14.4	–	15.3	–	12.8	–	–	–
Vision Banana (Gabeur et al., 2026)	✓	15.549	9.300	17.778	8.876	13.818	11.556	15.052	7.468	29.063	10.699
Task-specific LoRA	✓	16.781	9.341	18.39	8.578	16.997	12.09	14.957	7.355	28.213	12.221
Multi-task LoRA	✓	16.698	9.379	18.11	8.675	17.385	12.403	14.6	7.089	30.664	15.638
ReVision	✓	**15.467**	**9.162**	**16.202**	**7.994**	16.711	12.85	**13.489**	**6.644**	**24.854**	**8.43**

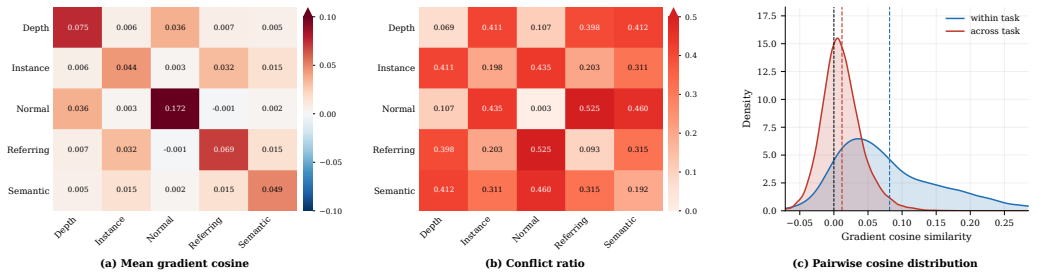
results on NYUv2, ETH3D, ScanNet, and iBims-1. Task-specific LoRA is competitive on several indoor datasets but degrades on outdoor datasets such as KITTI and DIODE, whereas Multi-task LoRA shows better cross-domain robustness. ReVision further improves on both, combining strong in-domain accuracy with better transfer across scene distributions.

Surface Normal Estimation. We evaluate surface normals using mean and median angular errors. As shown in Tab. 5, ReVision achieves strong zero-shot performance across both indoor and outdoor datasets, with particularly low errors on NYUv2, ScanNet, and VKitti. It improves the mean error over both LoRA baselines on nearly all datasets and achieves the best mean and median errors among our variants on VKitti. Compared with Multi-task LoRA, the gains are most pronounced under domain shift, indicating more robust transfer across scene distributions.

4.2 Analyses and Discussions

The key premise of ReVision is that perception should be represented by a conditional family of parameter updates rather than a single shared parameter point. To examine whether this structure emerges in practice, we analyze the LoRA updates generated for five training tasks: semantic segmentation, instance segmentation, referring segmentation, depth estimation, and surface-normal estimation. We sample 120 inputs per task, yielding 600 updates in total, and study them from three perspectives: task-level behavior, gradient interactions, and parameter-space geometry.

From Static Adaptation to Conditional Updates. Tabs. 1–5 compare ReVision with Task-specific LoRA and Multi-task LoRA across five perception tasks. On segmentation, Task-specific LoRA generally outperforms Multi-task LoRA, indicating interference from a single shared update across heterogeneous objectives. In contrast, Multi-task LoRA performs better on several geometric benchmarks and transfers more robustly under distribution shift, suggesting that joint training can capture useful cross-task structure. ReVision improves over both baselines across most settings, with clear gains on segmentation and more stable performance on depth and surface normals. These results highlight the limitation of static updates: task-specific LoRA specializes one parameter set per task, while multi-task LoRA shares one across all tasks. ReVision instead generates a distinct parameter update for each image–instruction pair, allowing different samples to occupy different points in weight space while retaining the benefits of joint multi-task training.



(a) Mean gradient cosine (b) Conflict ratio (c) Pairwise cosine distribution

Figure 3: **Gradient geometry across perception tasks.** **(a)** Mean pairwise cosine similarity between gradients of the generated LoRA parameters. **(b)** Conflict ratio, defined as the fraction of gradient pairs with negative cosine similarity. **(c)** Cosine-similarity distributions for within-task and cross-task pairs. Dashed lines mark zero and the distribution means.

Table 6: **Condition-routing ablation.** Only the parameter-generator condition is perturbed; image-generator inputs remain unchanged. We shuffle text or images within a task to break sample-level correspondence, and route conditions across tasks to break task-level correspondence. SA-Co/Gold, depth, and normals are omitted due to similar text conditions.

Method	Cityscapes			SA-Co/Gold	RefCOCOg	Depth		Normal	
	mIoU ↑	mAcc ↑	pixAcc ↑	pmF_1 ↑	cIoU ↑	AbsRel ↓	δ_1 ↑	mean° ↓	med.° ↓
Base Image Generator	20.29	36.79	59.91	4.32	7.64	0.5522	0.4662	105.84	109.21
Shuffled Text (within task)	67.74	79.19	94.20	-	46.9	-	-	-	-
Shuffled Image (within task)	68.03	79.34	94.21	48.5	51.7	0.062	0.950	18.83	11.83
Shuffled condition (cross task)	22.16	42.65	51.71	19.6	15.7	5.85	0.286	93.61	96.00
ReVision (Ours)	**70.27**	**81.36**	**94.79**	**51.1**	**54.2**	**0.049**	**0.971**	**17.99**	**11.02**

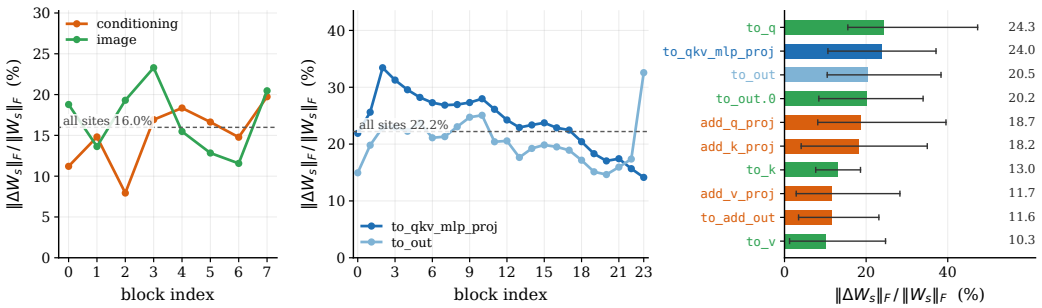
Cross-task Conflict in Gradient Space. We next ask whether the limitations of static multi-task updates are reflected in optimization geometry. We fix the flow timestep and noise realization, treat the ReVision-generated (A, B) as differentiable leaf tensors, and compute loss gradients with respect to these generated LoRA parameters after one flow step, without updating the backbone or parameter generator. As shown in Fig. 3, the mean cosine similarity drops from 0.082 within tasks to 0.012 across tasks, while the conflict ratio rises from 11.1% to 35.8%. The degree of interference is task dependent: depth and surface normals remain relatively aligned (0.036 cosine, 10.7% conflict), as do instance and referring segmentation (0.032, 20.3%), whereas several cross-family pairs conflict substantially more. Referring segmentation and surface normals, for example, reach a 52.5% conflict ratio. These results explain why a single multi-task update can become a compromise: it must reconcile gradients that vary substantially across tasks. ReVision instead generates a distinct update for each image–instruction instance, avoiding the need to collapse these directions into one shared parameter solution.

Task Structure with Instance-level Specialization. We examine the organization of generated updates in parameter space. Each LoRA update is represented by a 7,168-dimensional sketch and visualized with t-SNE. In Fig. 4(a), coloring the embedding by independent CLIP image-semantic clusters yields little separation (silhouette $= -0.006$). In contrast, task labels form well-separated regions (silhouette $= 0.839$; Fig. 4(b)), indicating that the coarse geometry is primarily task-structured. Task identity alone, however, does not determine the update. As shown in Tab. 6, breaking image–instruction correspondence within a task degrades performance, while cross-task routing causes substantially larger failures. This suggests a two-level organization: task determines the coarse region of parameter space, while the conditioning input selects an instance-specific update within that region.



(a) Colored by CLIP image semantics ($K = 7$) silhouette = -0.006 (b) Colored by task silhouette = 0.839

Figure 4: **Generated updates in parameter space.** The same t-SNE embedding of 7,168-dimensional LoRA update sketches is colored by (a) CLIP image-semantic clusters ($K = 7$) and (b) task identity. Silhouette scores are shown below each panel.



(a) Double-stream blocks **(b) Single-stream blocks** **(c) LoRA update magnitude**

Figure 5: **Layerwise and module-wise profile of the generated LoRA updates.** Relative update magnitude $\|\Delta W_s\|_F / \|W_s\|_F$ across **(a)** double-stream and **(b)** single-stream blocks, with dashed lines indicating the all-site mean. **(c)** Mean relative update magnitude for each projection type.

Geometry and Backbone Allocation of Generated Updates: (1) Task-structured update geometry. Tab. 7 shows clear task-dependent geometry in the generated updates. Within-task dispersion (0.024–0.281) is much smaller than that across all samples (0.573), while task centroids lie 2.5–14.8 within-task spreads from the global centroid, indicating distinct task-specific regions. Segmentation updates are particularly compact, whereas depth and surface normals exhibit greater variation. The effective rank and relative update magnitude remain stable across tasks, suggesting that task specialization is expressed primarily through the geometry of the generated updates rather than changes in their rank or scale. **(2) Structured allocation across the backbone.** Fig. 5 further shows that the generated updates are not distributed uniformly throughout the frozen generator. In the single-stream blocks, `to_qkv_mlp_proj` receives larger updates in early and middle layers and gradually decreases toward later layers, whereas `to_out` follows a distinct profile with a sharp increase in the final block. Double-stream blocks also exhibit different layerwise patterns between the conditioning and image branches. Across projection types, `to_q` and `to_qkv_mlp_proj` receive the largest relative updates, while value-related projections are consistently smaller. These patterns reveal structured allocation across both network depth and projection type, suggesting that the parameter generator selectively modulates different components of the frozen backbone rather than applying uniform adaptation.

Table 7: **Geometry of the generated update family. Centroid norm** measures the magnitude of the mean update, and **mean distance** measures the spread around it. Their ratio defines **dispersion**, the normalized update spread. **Offset/spread** measures task separation from the global centroid relative to within-task variation, while **participation ratio** estimates the effective dimensionality. We also report the effective rank and relative Frobenius norm $\|\Delta W_s\|_F / \|W_s\|_F$ of the generated LoRA updates. Detailed definitions are provided in the Appendix E.

	Sem.	Inst.	Ref.	Depth	Normal	All
Centroid norm	359.9	380.2	392.8	352.7	330.2	316.9
Mean distance	8.6	25.1	13.4	75.2	92.6	181.6
Dispersion	0.024	0.066	0.034	0.213	0.281	0.573
Offset / spread	14.8	5.5	11.9	2.8	2.5	–
Participation ratio	2.12	3.08	5.85	2.48	1.84	1.91
Effective rank (mean)	13.20	13.20	13.14	13.61	13.65	13.36
Effective rank (min)	3.72	3.99	4.04	3.68	3.52	3.52
Effective rank (max)	15.94	15.95	15.95	15.95	15.95	15.95
$\|\Delta W_s\|_F / \|W_s\|_F$ (mean)	18.54	19.53	19.97	18.04	17.19	18.65
$\|\Delta W_s\|_F / \|W_s\|_F$ (min)	1.64	1.73	1.80	1.34	1.24	1.24
$\|\Delta W_s\|_F / \|W_s\|_F$ (max)	37.39	40.22	39.56	44.24	47.32	47.32

5 Conclusion

We present ReVision, an instruction-conditioned parameter generation framework that repurposes a pretrained image generator for dense visual perception while keeping both the multimodal understanding model and generative backbone frozen. Given an image–instruction pair, ReVision maps its multimodal representation to transient low-rank updates that specialize the frozen generator for the requested prediction. This design enables different predictor behaviors to be instantiated directly from natural-language instructions without task-specific heads, explicit task identifiers, or stored adapters. Through a unified RGB interface, ReVision supports diverse dense prediction tasks, including segmentation, depth estimation, and surface-normal estimation. Our analyses further show that the generated updates exhibit clear task structure while retaining input-dependent variation, highlighting parameter generation as a flexible mechanism for adapting frozen generative backbones to heterogeneous perception objectives.

AI USE STATEMENT

Following the conference guidelines regarding generative artificial intelligence models (GenAI), we declare that GenAI was used only to assist with literature search, code development, and manuscript drafting and polishing. The research idea, method design, experimental methodology, data analysis, and core scientific content were developed and validated by the authors.

ETHICS STATEMENT

This work studies instruction-conditioned parameter generation for dense visual perception using frozen multimodal understanding and image-generation models. Our experiments use publicly available datasets and pretrained models. As with other large-scale vision and generative systems, ReVision may inherit biases and failure modes from the underlying models and training data. We do not identify additional ethical concerns specific to the proposed method beyond those commonly associated with visual perception and generative models.

REPRODUCIBILITY STATEMENT

We use publicly available datasets for training and evaluation and plan to release the code and implementation details to reproduce our results. The manuscript and appendix describe the model architecture, data construction, training procedure, hyperparameters, and evaluation protocols. Numerical results and tables were manually checked and cross-verified against experiment records. The authors take responsibility for the results, claims, and conclusions of this work.

REFERENCES

Roman Bachmann, Oğuzhan Fatih Kar, David Mizrahi, Ali Garjani, Mingfei Gao, David Griffiths, Jiaming Hu, Afshin Dehghan, and Amir Zamir. 4m-21: An any-to-any vision model for tens of tasks and modalities. In *NeurIPS*, 2024.

Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *CVPR*, 2024.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Rujikorn Charakorn, Edoardo Cetin, Yujin Tang, and Robert Tjarko Lange. Text-to-lora: Instant transformer adaption. In *ICML*, 2025.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607, 2020.

Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *ICLR*, 2022.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamal Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, 2023.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

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. *arXiv preprint arXiv:2010.11929*, 2020.

Zhongjie Duan and Yingda Chen. Compressing image style training into a single model forward. *arXiv preprint arXiv:2606.13809*, 2026.

Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *ICCV*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Shenghao Fu, Qize Yang, Qijie Mo, Junkai Yan, Xihan Wei, Jingke Meng, Xiaohua Xie, and Wei-Shi Zheng. Llmdet: Learning strong open-vocabulary object detectors under the supervision of large language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14987–14997. IEEE, 2025a.

Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *ECCV*, 2024.

Yunxiang Fu, Meng Lou, and Yizhou Yu. Segman: Omni-scale context modeling with state space models and local attention for semantic segmentation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19077–19087. IEEE, 2025b.

Valentin Gabeur, Shangbang Long, Songyou Peng, Paul Voigtlaender, Shuyang Sun, Yanan Bao, Karen Truong, Jonathan T Barron, Kyle Genova, Saining Xie, et al. Image generators are generalist vision learners. *arXiv preprint arXiv:2604.20329*, 2026.

Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, et al. Seedream 3.0 technical report. *arXiv preprint arXiv:2504.11346*, 2025.

Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *CVPR*, 2024.

Agrim Gupta, Piotr Dollár, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

David Ha, Andrew M Dai, and Quoc V Le. Hypernetworks. In *ICLR*, 2017.

Xiaoyang Han, Jianhua Li, Kewang Deng, Zukai Chen, Xuanke Shi, Sihan Wang, Zhongang Cai, Ziwei Liu, Lei Yang, Dahua Lin, et al. Vision as unified multimodal generation. *arXiv preprint arXiv:2607.06560*, 2026.

Jing He, Haodong Li, Mingzhi Sheng, and Ying-Cong Chen. Lotus-2: Advancing geometric dense prediction with powerful image generative model. *arXiv preprint arXiv:2512.01030*, 2025a.

Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. In *ICLR*, 2025b.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *NeurIPS*, 2016.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9339–9350. IEEE, 2025.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *ACL*, 2021.

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024.

Om Khangaonkar and Hamed Pirsiavash. Gen2seg: Generative models enable generalizable instance segmentation. *arXiv preprint arXiv:2505.15263*, 2025.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Miaoge Li, Yang Chen, Zhijie Rao, Can Jiang, and Jingcai Guo. Sg-lora: Semantic-guided lora parameters generation. In *CVPR*, 2026.

Zhiyuan Liang, Dongwen Tang, Yuhao Zhou, Xuanlei Zhao, Mingjia Shi, Wangbo Zhao, Zekai Li, Peihao Wang, Konstantin Schürholt, Damian Borth, Michael M Bronstein, Yang You, Zhangyang Wang, and Kai Wang. Drag-and-drop llms: Zero-shot prompt-to-weights. In *NeurIPS*, 2025.

Jiaxiang Liu, Zhida Feng, Pengyu Zou, Zhenyu Qian, Tianrui Zhu, Jun Xia, Yuehu Dong, Yanzheng Lin, Honglin Xiong, Anqi Chen, Yunpeng Ding, Jinghui Duan, Lin Gao, Chao Han, Tiechao He, Jiakang Hu, Ranjun Hua, Xueming Jiang, Qingli Kong, et al. ERNIE-Image Technical Report. *arXiv preprint arXiv:2605.25347*, 2026a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Ting Liu and Siyuan Li. Hybrid global-local representation with augmented spatial guidance for zero-shot referring image segmentation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29634–29643. IEEE, 2025.

Wei Liu, Jiaxin Lin, and Rui Chen. Open-source image editing models are zero-shot vision learners. *arXiv preprint arXiv:2605.04566*, 2026b.

Yen-Cheng Liu, Chih-Yao Ma, Junjiao Tian, Zijian He, and Zsolt Kira. Polyhistor: Parameter-efficient multi-task adaptation for dense vision tasks. In *NeurIPS*, 2022.

Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *ICLR*, 2023.

Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. In *CVPR*, 2024.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023.

David Mizrahi, Roman Bachmann, Oğuzhan Fatih Kar, Teresa Yeo, Mingfei Gao, Afshin Dehghan, and Amir Zamir. 4m: Massively multimodal masked modeling. In *NeurIPS*, 2023.

Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017.

Yuval Nirkin, Lior Wolf, and Tal Hassner. Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In *CVPR*, 2021.

Jason Phang, Yi Mao, Pengcheng He, and Weizhu Chen. Hypertuning: Toward adapting large language models without back-propagation. In *ICML*, 2023.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763, 2021.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13009–13018. IEEE, 2024.

Tianhe Ren, Yihao Chen, Qing Jiang, Zhaoyang Zeng, Yuda Xiong, Wenlong Liu, Zhengyu Ma, Junyi Shen, Yuan Gao, Xiaoke Jiang, et al. Dino-x: A unified vision model for open-world object detection and understanding. *arXiv preprint arXiv:2411.14347*, 2024.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models. In *CVPR*, 2024.

Yunhang Shen, Chaoyou Fu, Peixian Chen, Mengdan Zhang, Ke Li, Xing Sun, Yunsheng Wu, Shaohui Lin, and Rongrong Ji. Aligning and prompting everything all at once for universal visual perception. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13193–13203. IEEE, 2024.

Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. In *ICLR*, 2024.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *NeurIPS*, 2023.

Tencent HY Team. Hy-wu (part i): An extensible functional neural memory framework and an instantiation in text-guided image editing. *arXiv preprint arXiv:2603.07236*, 2026.

Bowen Tian, Wenshuo Chen, Zexi Li, Songning Lai, Jiemin Wu, and Yutao Yue. Text2weight: Bridging natural language and neural network weight spaces. In *ACM MM*, 2025.

Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020.

Hao Wang, Limeng Qiao, Zequn Jie, Zhijian Huang, Chengjian Feng, Qingfang Zheng, Lin Ma, Xiangyuan Lan, and Xiaodan Liang. X-sam: From segment anything to any segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26187–26196, 2026.

JiYuan Wang, Chunyu Lin, Lei Sun, Rongying Liu, Lang Nie, Mingxing Li, Kang Liao, and Xiangxiang Chu. From editor to dense geometry estimator. *arXiv preprint arXiv:2509.04338*, 2025.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *IROS*, 2020a.

Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *NeurIPS*, 2020b.

Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Towards segmenting everything in context. In *ICCV*, 2023.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Cong Wei, Yujie Zhong, Haoxian Tan, Yong Liu, Zheng Zhao, Jie Hu, and Yujiu Yang. Hyperseg: Towards universal visual segmentation with large language model. *arXiv preprint arXiv:2411.17606*, 2024.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *CVPR*, 2025.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *ICLR*, 2025.

Guangkai Xu, Yongtao Ge, Mingyu Liu, Chengxiang Fan, Kangyang Xie, Zhiyue Zhao, Hao Chen, and Chunhua Shen. What matters when repurposing diffusion models for general dense perception tasks? In *ICLR*, 2025a.

Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, 2023.

Yifeng Xu, Zhenliang He, Meina Kan, Shiguang Shan, and Xilin Chen. Jodi: Unification of visual generation and understanding via joint modeling. *arXiv preprint arXiv:2505.19084*, 2025b.

Timing Yang, Jinrui Yang, Xinlong Li, Yuhan Wang, Haoran Li, Chen Wei, Rama Chellappa, Yuyin Zhou, Cihang Xie, Alan Yuille, et al. Let rgb be the language of vision. *arXiv preprint arXiv:2607.12450*, 2026a.

Yuhuan Yang, Xianwei Zhuang, Yuxuan Cai, Chaofan Ma, Shuai Bai, Jiangchao Yao, Ya Zhang, Junyang Lin, and Yanfeng Wang. Genmask: Adapting dit for segmentation via direct mask generation. *arXiv preprint arXiv:2603.23906*, 2026b.

Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. Stablenormal: Reducing diffusion variance for stable and sharp normal. *ACM Transactions on Graphics*, 2024.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020.

Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1020–1031. IEEE, 2023.

Jianpeng Zhang, Yutong Xie, Yong Xia, and Chunhua Shen. Dodnet: Learning to segment multi-organ and tumors from multiple partially labeled datasets. In *CVPR*, 2021a.

Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. In *NeurIPS*, 2021b.

Bing Zhao, Chenfei Wu, Deqing Li, Hao Meng, Jiahao Li, Jie Zhang, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kuan Cao, et al. Qwen-image-2.0 technical report. *arXiv preprint arXiv:2605.10730*, 2026a.

Canyu Zhao, Yanlong Sun, Mingyu Liu, Huanyi Zheng, Muzhi Zhu, Zhiyue Zhao, Hao Chen, Tong He, and Chunhua Shen. Diception: A generalist diffusion model for visual perceptual tasks. *Advances in Neural Information Processing Systems*, 38:40108–40155, 2026b.

Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *ICCV*, 2023.

Zuyan Zhao, Zhenliang He, Meina Kan, Shiguang Shan, and Xilin Chen. Unipercept: A unified diffusion model for generalizable visual perception. In *CVPR*, 2026c.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *ICLR*, 2025.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15116–15127. IEEE, 2023.

APPENDIX

A IMPLEMENTATION DETAILS

In-context conditioning. The reference image and prediction target are independently encoded by the frozen VAE, patchified, and normalized using the native latent statistics of the image generator. Following the multi-latent formulation of `FLUX.2-klein-base-9B`, the target and reference tokens are concatenated into a single sequence and assigned different temporal coordinates, with $T = 0$ for the prediction target and $T = 10$ for the reference image. The remaining spatial coordinates follow their native patch positions. The flow-matching loss is computed only over the target-token portion. The parameter generator receives only the multimodal representation of the reference image and instruction and never observes the noisy target.

	stage 1	stage 2
resolution	256×256	512×512
steps	10 000	20 000
micro-batch / GPU	24	8
gradient accumulation	1	1
optimizer	AdamW, $\beta = (0.9, 0.999)$, weight decay 10^{-4}	
peak learning rate	2×10^{-5}	2×10^{-5}
warmup steps	500	500
schedule	warmup $\rightarrow$ constant	
gradient clipping	1.0	
precision	`bf16`	
EMA	decay 0.999, no warmup	
timestep density	$t = \sigma(u),\ u \sim \mathcal{N}(0, 1)$	

Table 8: **Training configuration.** The stage transition loads *only* the ReVision weights: global step, optimizer state, learning-rate schedule, EMA and data-order RNG are all reset, so stage 2 performs a full warmup to the same peak learning rate rather than inheriting stage 1's step counter.

Flow matching and timestep sampling. Let $\mathbf{e}^{\mathrm{gt}}$ denote the encoded target and $\varepsilon \sim \mathcal{N}(0, I)$. We construct

$$\mathbf{x}_t = (1 - t)\mathbf{e}^{\mathrm{gt}} + t\varepsilon, \qquad \mathbf{v} = \varepsilon - \mathbf{e}^{\mathrm{gt}}. \tag{8}$$

We sample t from a standard logit-normal distribution and apply a fixed resolution-independent shift of $e^{0.8} \approx 2.2255$. The generated parameters depend only on the image–instruction pair and are computed once per sample, remaining fixed throughout the denoising trajectory.

Optimization and multi-task training. Training proceeds in two stages. We first train at 256^2 for 10k steps with 500 warmup steps on semantic segmentation, instance segmentation, depth estimation, and surface-normal estimation. We then transfer the raw non-EMA parameter-generator weights to the 512^2 stage and train for another 20k steps with 500 warmup steps, additionally introducing referring segmentation. The optimizer and learning-rate schedule are restarted between stages. We use AdamW with a peak learning rate of 2.5×10^{-5} and a warmup-stable schedule without terminal decay. At each optimization step, we draw an equal number of samples from every task-specific dataloader and concatenate them into a mixed-task batch. Thus, all tasks contribute equally to each update and are processed jointly in a single batched forward pass through the frozen backbones. We maintain an EMA of the parameter-generator weights with decay 0.999 and use the EMA checkpoint for evaluation. Detailed hyperparameters are summarized in Tab. 8.

Zero initialization. To stabilize optimization, we use standard initialization for all modules except the output projections that generate the LoRA B matrices, which are initialized to zero. As a result, $B_s = 0$ and therefore $\Delta W_s = \frac{\alpha}{r} B_s A_s = 0$ at every injection site at initialization, regardless of the initial values of A_s. The adapted generator is thus functionally identical to the pretrained

generator at the start of training. This avoids introducing arbitrary parameter perturbations before the conditioning pathway has learned a meaningful mapping from multimodal representations to parameter updates. During optimization, the parameter generator gradually departs from this zero-update state and learns input-dependent low-rank modifications while the generator remains fixed.

B Unified RGB Output and Data Generation

Unified data interface. We cast all perception tasks into the same image-to-image training format. Each training sample consists of an RGB input image, a natural-language instruction, and an RGB target image. Task-specific annotations are converted into RGB representations before training, with reserved colors used for invalid or background pixels when applicable. Input images are resized with standard image interpolation, whereas discrete annotations are always resized with nearest-neighbor interpolation to avoid introducing spurious class or instance labels. Samples are resized when necessary and cropped into fixed-resolution square regions without padding. This standardizes spatial dimensions across tasks and ensures that both the input and target satisfy the image generator's resolution constraint that height and width be multiples of 16. The resulting unified interface allows semantic segmentation, instance segmentation, referring segmentation, depth estimation, and surface-normal estimation to share the same generative training objective.

Semantic segmentation. Semantic annotations are first converted into class-index maps and then rendered as RGB targets using a sample-specific color palette. For each crop, foreground classes are assigned distinct colors sampled without replacement, while background and ignored regions are mapped to black. If more than 32 classes occur in a sample, we retain the classes with the largest pixel areas and map the remaining regions to background. The resulting class-to-color mapping is optionally included in the instruction. Because the palette is redrawn independently for each sample, the model cannot associate a semantic class with a fixed dataset-level color. We further randomize how colors are expressed in language, RGB, and hexadecimal codes. A crop is discarded if it contains only ignored pixels and therefore provides no valid semantic supervision.

Instance segmentation. For instance segmentation, we use the original annotations primarily to identify the semantic category and the expected object support, while regenerating high-quality instance masks with SAM 3 (Carion et al., 2026) using the category name as the text prompt. This avoids directly supervising the model with coarse or noisy source masks. Each training row corresponds to one image-category pair, and different instances of the queried category are rendered with different colors. We apply several stages of quality control before constructing the RGB target. First, SAM 3 predictions must satisfy a minimum confidence of 0.55, and masks with more than 25% of their pixels in the uncertain probability range $(0.3, 0.7)$ are rejected. We then remove masks with extreme area ratios or aspect ratios, low mask-to-box occupancy, strong overlap with higher-confidence predictions, or excessive fragmentation. Specifically, valid masks must occupy between 0.0015 and 0.92 of the image, have a box aspect ratio below 12, and fill at least 20% of their bounding box. Mask-level non-maximum suppression is applied at an IoU threshold of 0.7, and highly fragmented predictions are removed. We additionally compare the union of the regenerated masks with the original annotation. Image-category pairs are rejected when their union IoU or recall falls below dataset-specific thresholds, which prevents incorrect SAM 3 detections from entering the training set. Crops are then constructed around the surviving objects. We include a context crop centered on the union of all instances and up to three object-centric crops around the largest instances. Highly redundant crops with box IoU above 0.85 are removed. Instances for which less than 25% of the original mask remains inside a crop are treated as background rather than as valid truncated objects. Each surviving instance is assigned a distinct color, with colors selected to be well separated in perceptual RGB space. When the instruction explicitly specifies the background color, it is included in the same separation constraint. After resizing the crop to the final training resolution, we perform a final consistency check and reject samples in which small instances disappear or the number of visible instance colors no longer agrees with the instruction. This ensures that the textual description and rendered supervision remain synchronized.

Referring segmentation. Referring segmentation uses the same SAM 3 (Carion et al., 2026)-based mask regeneration and object-centric cropping pipeline, but the segmentation prompt is the referring expression itself rather than a category name. We first sanitize the text before running

SAM 3. Truncated expressions containing ellipses, dangling function words, purely deictic phrases, excessively long descriptions, and duplicated expressions are removed. We further require each cleaned expression to correspond to a unique object identity. After SAM 3 filtering, an expression is retained only if exactly one valid instance remains. This provides an additional visual uniqueness check beyond the source annotation. Crops must also preserve a sufficient fraction of the referred object, since a strongly truncated target is no longer a faithful visual grounding of the original expression. The target is represented as a two-color RGB mask, with the referred object assigned a foreground color and all remaining pixels assigned a background color. Both colors may be specified in the instruction and are used directly to render the target. Images appearing in the SA-Co/Gold evaluation set are excluded from this training source to prevent leakage into the zero-shot evaluation.

Relative depth estimation. We encode relative depth as an RGB image using a continuous one-dimensional color code Γ defined along a Hamiltonian path through the eight vertices of the RGB cube. For a normalized scalar $\rho \in [0, 1]$, we use

$$\Gamma(\rho) = v_j + (7\rho - j)(v_{j+1} - v_j), \qquad j = \lfloor 7\rho \rfloor, \tag{9}$$

where adjacent vertices differ in only one RGB channel. This construction gives a continuous and invertible mapping between a scalar relative depth value and an RGB target. Normalization is performed independently for each crop rather than for the entire source image. Let ℓ and h denote the 2nd and 98th percentiles of valid depth values within the crop. We compute

$$u = \mathrm{clip}\left(\frac{x - \ell}{h - \ell}, 0, 1\right), \qquad \rho = \tau + (1 - \tau)u, \tag{10}$$

where the orientation of x is chosen such that u increases with distance. We use $\tau = 0.005$ to reserve pure black $(0, 0, 0)$ exclusively for invalid pixels. The per-image normalization is important because the model observes only the cropped image during training; using statistics from regions outside the crop would introduce an unobservable dependence into the target. We remove crops with insufficient geometric supervision. A crop is discarded when valid depth occupies less than 20% of its pixels. We also remove crops whose robust relative depth variation satisfies

$$\frac{h - \ell}{\max(|h|, |\ell|)} < 0.01. \tag{11}$$

This prevents nearly planar or constant-depth regions from being stretched across the full color range, which would amplify small sensor or reconstruction noise into artificial supervision. Since this representation intentionally removes absolute scale and shift, predictions are aligned to ground truth with a least-squares scale-and-shift transformation during evaluation.

Surface-normal estimation. Surface normals are represented directly as RGB images by mapping the three components of a unit normal from $[-1, 1]$ to $[0, 255]$. Predicted RGB values are mapped back to $[-1, 1]$ and renormalized to unit length, while invalid pixels are represented by black. Normals from different sources are converted to a common camera coordinate convention with X_{cam} pointing right, Y_{cam} pointing down, and Z_{cam} pointing into the scene. For sources such as TartanAir (Wang et al., 2020a) that provide depth but not surface-normal labels, we derive normals directly from depth. Each pixel is back-projected into camera-space 3D using the camera intrinsics, and a local plane is fitted within a 5×5 neighborhood. Neighbors whose relative depth differs from the center pixel by more than 3% are excluded to avoid fitting across occlusion boundaries. The normal is obtained from the eigenvector associated with the smallest eigenvalue of the local covariance matrix and is oriented consistently toward the camera. Pixels with invalid depth or insufficient valid neighbors are marked as invalid. Unlike relative depth, normal targets do not require per-crop scale normalization, and valid crops are retained without an additional global quality threshold.

Instruction construction. Instructions are sampled from hand-written task-specific templates. For geometry tasks, templates provide different natural-language formulations of the same prediction objective, with the depth instructions explicitly using relative or normalized terminology rather than metric-depth language. For segmentation tasks, instructions can additionally contain the class-to-color, instance-color, or foreground-background mapping used to construct the target. Semantic segmentation includes the color mapping with probability 0.8, while a fraction of the remaining samples contain only the task instruction. Most importantly, the colors stated in an instruction are

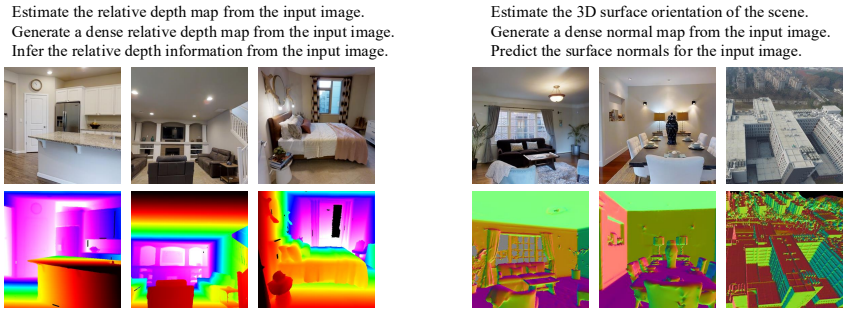
Table 9: **Training data sources.** Datasets and sample counts used in the 256^2 and 512^2 training stages.

Task	Training sources	256^2	512^2
Semantic seg.	ADE20K, COCO-Stuff-164K, Mapillary Vistas v2, BDD100K	0.69M	1.0M
Instance seg.	LVIS, Open Images V7, GranD	2.28M	7.6M
Referring seg.	GranD	0	3.0M
Relative depth	Omnidata (Hypersim, BlendMVS, Replica, Replica-GSO, Taskonomy), TartanAir	1.7M	2.7M
Surface normals	Omnidata (Hypersim, BlendMVS, Replica, Replica-GSO, Taskonomy), TartanAir	1.8M	2.9M
Total		6.47M	17.2M



(a) Semantic segmentation (b) Instance segmentation (c) Referring segmentation

Figure 6: **Examples from the datasets used for different segmentation tasks.**



(a) Depth estimation (b) Surface normal estimation

Figure 7: **Examples from the depth and surface-normal datasets.**

always the exact colors used to render the corresponding target pixels. This prompt-target consistency is enforced after all cropping and resizing operations and forms a final quality-control criterion for the segmentation datasets.

The data sources for all tasks are summarized in Tab. 9, with representative examples visualized in Fig. 6 and Fig. 7.

C Model Details

LoRA injection sites. We use `FLUX.2-klein-base-9B` (Labs, 2025) as the frozen base generator. It contains 8 double-stream blocks followed by 24 single-stream blocks. The parameter generator predicts LoRA parameters for the attention-related linear projections in the backbone. Specifically, each double-stream block contains eight injection sites: `to_q`, `to_k`, `to_v`, and `to_out.0` for the image stream, and `add_q_proj`, `add_k_proj`, `add_v_proj`, and `to_add_out` for the conditioning stream. Each single-stream block contains two injection sites, `to_qkv_mlp_proj` and `to_out`. This gives a total of $8 \times 8 + 24 \times 2 = 112$ LoRA injection sites. For each site, the parameter generator produces the corresponding LoRA-A and LoRA-B matrices, while the base-model parameters remain frozen.

ReVision parameter budget. The parameter generator consists of an instruction bridge, a learnable query bank, $N = 10$ Transformer blocks, and site-specific decode heads that generate the LoRA factors for the frozen image generator. We use hidden width $d = 1536$, 24 attention heads, bottleneck width $m = 256$, and LoRA rank $r = 16$. For $S = 112$ injection sites, the query bank contains $2Sr = 3584$ learnable tokens. A detailed breakdown of the parameter budget is provided in Tab. 10.

Table 10: **Parameter budget of the ReVision parameter generator.** Parameter counts omit linear-layer biases. The per-site decode heads dominate the budget because they produce LoRA factors at the native widths of the generator backbone.

Module	Shape	Params
Instruction bridge	$\mathrm{Linear}(4096 \to 1536)$	6.3M
Learnable queries Q	3584×1536	5.5M
Transformer blocks	$10 \times [\mathrm{Linear}(d \to 9d) + \mathrm{Linear}(4d \to d)]$	306.7M
Per-site decode heads	$112 \times [2\,\mathrm{Linear}(d \to m) + \text{LoRA decoding}]$	599.8M
Total		**918.3M**

D Evaluation Protocols

All evaluations use 30 sampling steps, a guidance scale of 4.0, and a fixed random seed of 42.

Segmentation. We formulate these tasks as RGB mask generation and decode predictions by nearest-colour assignment over the palette specified in the instruction. For semantic segmentation, we evaluate all 500 Cityscapes `val` images at 1024×2048. The instruction specifies the semantic classes present in the image and their corresponding colours, and predictions are evaluated against the 19-class ground truth with `ignore_index`$= 255$. For instance segmentation, the instruction specifies the target object category, and the model predicts the corresponding instance mask with all other regions treated as background. For referring segmentation, the target is specified by a free-form referring expression, and the model predicts the mask of the referred region.

Depth. We evaluate affine-invariant monocular depth following the Marigold protocol. For each image, we generate a single RGB prediction and decode it into relative depth, without ensembling. We then fit a least-squares scale and shift in depth space to align the prediction with the ground truth, and clip the aligned depth to the valid range of each dataset. Evaluation uses the same resolutions, masks, and depth ranges as Marigold, with all metrics computed at the native ground-truth resolution.

Normals. We evaluate surface-normal estimation zero-shot on NYUv2, DIODE-indoor, ScanNet, and Virtual KITTI 2, without dataset-specific fine-tuning. The predicted RGB output is decoded into unit surface normals, transformed to the evaluation coordinate convention, and resized to the native ground-truth resolution before evaluation.

E Analysis Details

This section provides the definitions of the statistics used in section 4.2. Unless otherwise stated, all analyses are performed on the generated LoRA updates from the same trained parameter generator.

E.1 Parameter-Space Representation

Directly vectorizing the generated weight updates is prohibitively expensive. For an injection site s, the LoRA update is

$$\Delta W_s = \frac{\alpha}{r} B_s A_s. \tag{12}$$

We therefore represent each update using a fixed factorized random projection:

$$S_s = R_{1,s}^\top \Delta W_s R_{2,s} = \frac{\alpha}{r} \left(R_{1,s}^\top B_s\right) \left(A_s R_{2,s}\right), \tag{13}$$

where $R_{1,s}$ and $R_{2,s}$ are fixed Gaussian projection matrices. We use 8×8 projections at each of the 112 injection sites and concatenate their vectorized outputs:

$$z = \mathrm{Concat}_{s=1}^{112} \mathrm{vec}(S_s) \in \mathbb{R}^{7168}. \tag{14}$$

The same projections are shared across all samples, allowing distances and cosine similarities to be compared in a common parameter space.

E.2 Update Magnitude and Effective Rank

The Frobenius norm of each update can be computed directly from its low-rank factors without constructing ΔW_s:

$$\|\Delta W_s\|_F^2 = \left(\frac{\alpha}{r}\right)^2 \mathrm{tr}\left[(A_s A_s^\top)(B_s^\top B_s)\right]. \tag{15}$$

Since injection sites have different matrix sizes, comparisons across sites use the relative update magnitude

$$m_s = \frac{\|\Delta W_s\|_F}{\|W_s\|_F}. \tag{16}$$

We additionally measure the effective rank of each generated update. Let $\{\sigma_j\}$ denote its non-zero singular values and $p_j = \sigma_j / \sum_l \sigma_l$. The effective rank is

$$\mathrm{erank}(\Delta W_s) = \exp\left(-\sum_j p_j \log p_j\right). \tag{17}$$

It measures how broadly the available rank-r subspace is utilized and is bounded by r.

E.3 Geometry of Generated Updates

Let z_n denote the parameter sketch of sample n, and let $\mathcal{G}$ denote a group of samples, such as one task. Its **centroid** and **mean distance** to the centroid are

$$\mu_{\mathcal{G}} = \frac{1}{|\mathcal{G}|} \sum_{n \in \mathcal{G}} z_n, \qquad d_{\mathcal{G}} = \frac{1}{|\mathcal{G}|} \sum_{n \in \mathcal{G}} \|z_n - \mu_{\mathcal{G}}\|_2. \tag{18}$$

We report the normalized **dispersion**

$$D_{\mathcal{G}} = \frac{d_{\mathcal{G}}}{\|\mu_{\mathcal{G}}\|_2}, \tag{19}$$

and the **task-centroid offset** relative to its within-task spread:

$$O_{\mathcal{G}} = \frac{\|\mu_{\mathcal{G}} - \mu_{\mathrm{all}}\|_2}{d_{\mathcal{G}}}. \tag{20}$$

To characterize the dimensionality of the variation, we compute the **participation ratio**:

$$\mathrm{PR}(\mathcal{G}) = \frac{\left(\sum_j \lambda_j\right)^2}{\sum_j \lambda_j^2}, \tag{21}$$

where $\{\lambda_j\}$ are the eigenvalues of the covariance matrix of the centered sketches in $\mathcal{G}$. A smaller value indicates that the variation is concentrated along fewer dominant directions.

E.4 Gradient Conflict

For each training sample n, we compute the gradient of the flow-matching objective with respect to its generated LoRA parameters and represent it as g_n. For two samples a and b, their gradient similarity is measured by

$$C(a, b) = \frac{\langle g_a, g_b \rangle}{\|g_a\|_2 \|g_b\|_2}. \tag{22}$$

For a set of sample pairs $\mathcal{P}$, we report the mean cosine similarity and the conflict ratio

$$\mathrm{Conflict}(\mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{(a,b) \in \mathcal{P}} \mathbf{1}[C(a, b) < 0]. \tag{23}$$

We compute these statistics separately for within-task and cross-task pairs.

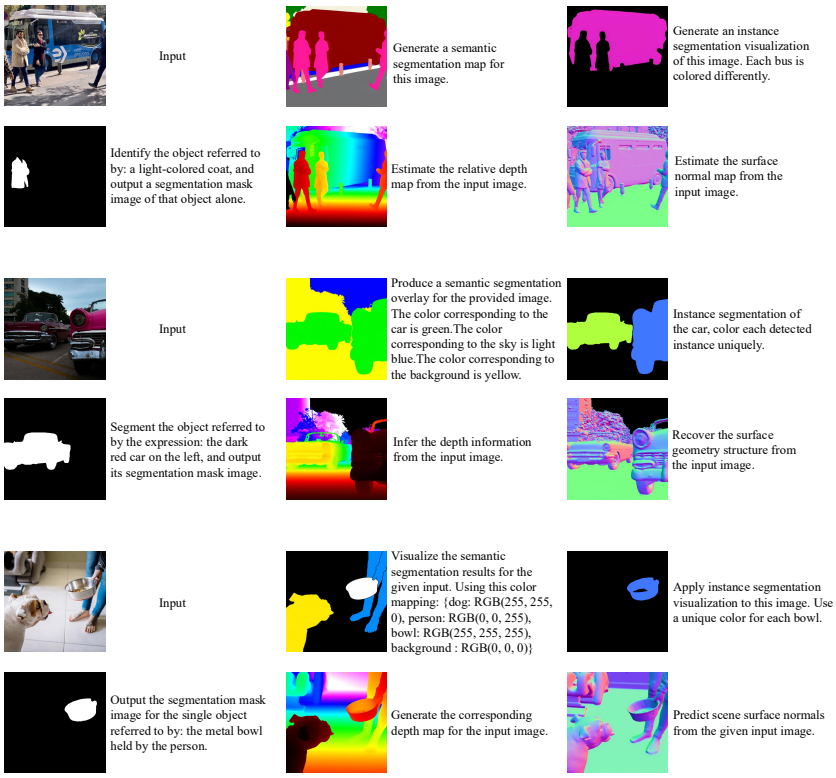


Figure 8: **Qualitative results across dense perception tasks.** ReVision follows free-form image–instruction inputs to produce pixel-space outputs for semantic segmentation, instance segmentation, referring segmentation, depth estimation, and surface-normal prediction. Different colors and output conventions are specified directly by the instruction.

E.5 Parameter-Space Separability

We use the silhouette score to measure the separation of generated updates by task identity or image-semantic clusters. For sample m, let a_m denote its mean distance to samples in the same cluster and dis_m the minimum mean distance to samples in any other cluster. We define

$$\text{silh}_m = \frac{dis_n - a_m}{\max(a_m, dis_m)}, \qquad \text{silhouette score} = \frac{1}{N}\sum_{n=1}^{N} \text{silh}_m. \tag{24}$$

All distances are computed in the parameter-sketch space, while t-SNE is used only for visualization.

F Visualization Results

Fig. 8 shows representative predictions across five dense perception tasks. Given the same input image, ReVision can instantiate different prediction behaviors according to the accompanying natural-language instruction. For semantic segmentation, the generated masks follow the color mapping specified in the prompt rather than relying on a fixed dataset-level palette. Instance segmentation separates individual objects with distinct colors, while referring segmentation localizes only the region described by the expression. For geometric prediction, the generated depth maps preserve the relative ordering of foreground and background regions, and the surface-normal predictions exhibit spatially coherent orientation changes along object and scene boundaries. These examples also illustrate that a shared RGB output space can support both discrete segmentation masks and continuous geometric predictions without task-specific

G Preserving Generation and Editing Capabilities

A key property of ReVision is that learning perception does not overwrite the pretrained generator. The image generator remains frozen throughout training, and the generated LoRA updates are transient and used only for the current image–instruction pair. When these updates are removed, ReVision exactly recovers the original generator parameters and can directly use its native image generation and editing capabilities. This differs from perception methods that fine-tune the generative backbone, where perceptual objectives and generative behavior must coexist within the same optimized parameters. In ReVision, generation and editing remain associated with the original pretrained model, while dense perception is instantiated through input-dependent parameter updates. As a result, adding perception does not require replaying generative data or explicitly balancing generation and perception objectives during training. Because the backbone parameters are never modified, its native generation and editing functions remain directly accessible by running the generator without the generated perception updates.

H Limitations

ReVision is still bounded by the capabilities of its frozen multimodal understanding model and image generator, since errors in either backbone cannot be corrected through backbone fine-tuning. Instance-level parameter generation also introduces additional computation compared with using a single fixed adapter. In addition, all dense predictions are expressed through RGB generation, which is flexible but may be less efficient than task-specific prediction heads and remains sensitive to the native generative training objective. This is particularly relevant for tasks that require precise masks or geometric predictions. Future work could explore stronger frozen backbones, more efficient parameter generators, and training objectives better aligned with dense prediction.