

GeoBridge: Evolving Transferable Decoder Topologies for Geometric Vision

Rui Wang, Haoyu Zhang, and Ran Cheng, *Senior Member, IEEE*

Abstract—Depth estimation and Gaussian reconstruction share a need for multi-scale feature processing, yet use different prediction interfaces and training objectives. Whether a decoder’s internal computation can be optimized once and reused across these systems remains a structural design question. Search over primitive configurations alone leaves the organization of transformations fixed. To address this gap, we propose GeoBridge, an evolutionary framework for transferable decoder topologies. First, GeoBridge makes stage-internal topology searchable through residual, sequential, and parallel composition of configurable primitives. Shape-preserving composition retains the stage interfaces of a fixed multi-scale scaffold. Second, a functional-tree encoding supports stage-aware crossover and hierarchical mutation, while a gradient sensitivity-alignment proxy evaluates candidates without training. Multi-objective evolution balances this proxy with computation and parameter cost. Third, selected topologies are instantiated and trained in target systems with task-specific prediction interfaces. A single source search takes approximately 12 GPU hours. GeoBridgeNet-A3 achieves the best depth accuracy among four adapted decoder baselines under a common encoder and training protocol on NYU Depth V2 and KITTI. The same topology reaches 27.11 dB PSNR in Gaussian novel-view synthesis on RealEstate10K, and its refined representations support open-vocabulary segmentation on LERF. These results support decoder topology as a reusable design object across the evaluated geometric vision systems. Code and pretrained weights are available at <https://github.com/EMI-Group/GeoBridge-NAS>.

Keywords: Decoder topology, architecture transfer, geometric vision, evolutionary neural architecture search, functional composition, training-free evaluation.

I. INTRODUCTION

DECODERS connect visual features to geometric predictions through multi-scale processing and spatial detail recovery. In monocular depth estimation (MDE), they map image features to dense depth [1]–[3]. In feed-forward Gaussian reconstruction, decoded features contribute to depth inference and the prediction of a renderable scene representation [4]–[6]. These systems have different prediction interfaces and training objectives, yet share a need to organize feature transformations across spatial scales. This common role motivates a structural question: can a decoder’s internal computation be optimized in one geometric setting and remain useful in another?

Answering this question requires treating the organization of decoder computation as a design variable. In a decoder

stage with a prescribed composition, changing a primitive’s depth, scan pattern, or fusion rule leaves the surrounding serial, parallel, and residual arrangement fixed. These arrangements determine how transformations accumulate along a path and how branch outputs are aggregated. Neural architecture search (NAS) has explored operation connectivity, multi-scale fusion, and hierarchical motifs [7]–[9]. For decoder reuse across geometric systems, the specific design problem is to expose stage-internal composition to search while retaining the interfaces needed by the surrounding network.

We investigate the hypothesis that decoder topology can serve as a reusable design object across geometric prediction systems. Here, topology specifies the residual, sequential, and parallel organization of transformations within each decoder stage. Transfer retains this organization and the primitive configurations; each target system trains its own weights and prediction interface. A fixed multi-scale scaffold supplies compatible stage boundaries, and variable internal topologies provide alternative ways to process the features at those boundaries. This separation permits one source search followed by target training under different prediction objectives.

To realize this design, we propose GeoBridge, an evolutionary framework that searches decoder topologies in a depth-estimation scaffold and reuses the selected structures in target geometric systems. A bounded composition grammar constructs each stage from shape-preserving primitives through residual, sequential, and parallel operators. Functional trees encode these compositions, so stage-aware crossover and hierarchical mutation can vary complete stages, subtrees, and primitive configurations. A gradient sensitivity-alignment proxy (GSAP) evaluates independently initialized candidates under a common teacher-derived regression stimulus. NSGA-II [10] balances this training-free score with computation and parameter cost to obtain a set of decoder structures for subsequent reuse.

Experiments test whether the structures selected in this source setting retain their utility after target training. One search on TartanAir [11] produces the GeoBridgeNet family in approximately 12 GPU hours. GeoBridgeNet-A3 achieves the best depth accuracy among four recent decoder baselines under a common encoder and training protocol on NYU Depth V2 and KITTI. The same topology reaches 27.11 dB PSNR after integration into a Gaussian novel-view synthesis (NVS) system on RealEstate10K. Its Gaussian outputs also support open-vocabulary segmentation on LERF after scene refinement and semantic optimization [12]–[14]. Controlled studies characterize the decoder designs, the structural coverage of evolutionary variation, and the ranking quality of GSAP.

Rui Wang and Haoyu Zhang are with the School of Information Science and Technology, Hangzhou Normal University, Hangzhou, P.R. China (e-mail: kujiyunmili@gmail.com; haoyu.zhang@hznu.edu.cn).

Ran Cheng is with the Department of Data Science and Artificial Intelligence, The Hong Kong Polytechnic University, Hong Kong SAR, China (e-mail: ranchengcn@gmail.com).

The main contributions of this work are summarized as follows:

- We formulate decoder design as evolution of stage-internal computation topologies within a fixed multi-scale scaffold. A shape-preserving composition grammar opens residual, sequential, and parallel organization to search alongside primitive configuration.
- We develop an evolutionary search over these compositions with stage-aware crossover, hierarchical mutation, and GSAP evaluation. This combination supports independent candidate instantiation and multi-objective selection without candidate training.
- We demonstrate reuse of the selected decoder structures across real-world depth estimation and Gaussian novel-view synthesis through target-specific training. The evaluation connects a single source search to prediction systems with different output representations and further examines the semantic utility of their reconstructed scenes.

Section II reviews decoder design, architecture transfer, structural encoding, and candidate evaluation. Section III presents the topology space, evolutionary search, and target instantiation. Section IV evaluates the resulting structures and their reuse, followed by conclusions in Section V.

II. RELATED WORK

The design of transferable decoder topologies connects architecture search with structural reuse. We first review how NAS exposes visual computation to optimization and how architectures are transferred between prediction tasks. We then examine the representations and evaluation methods that support evolutionary search over composed structures.

A. Architecture Search for Visual Decoding

NAS combines an architecture space, a search strategy, and a performance estimator to automate network design. Early studies focused on image classification and established reinforcement learning [15], gradient-based optimization [7], and evolutionary search [16] as major search strategies. Evolutionary NAS directly manipulates discrete operations and connections through selection, crossover, and mutation. Its population-based formulation also supports the joint optimization of prediction quality and computational cost.

Dense prediction has motivated architecture spaces tailored to multi-scale processing. NAS-FPN and FNA++ search feature fusion and network structures for object detection [8], [17]. Auto-DeepLab searches multi-resolution structures for semantic segmentation [18], while SCTNet-NAS incorporates deployment efficiency [19]. MDE-oriented methods further specialize search to depth prediction [20]–[22]. Task-specific search spaces also address feature extraction and fusion in other vision problems. TREFE searches the spatial and sequential feature extractors of text recognition networks under a latency constraint [23], and NAS-PED searches hybrid convolution–transformer backbones for pedestrian detection in crowded scenes [24]. SPOSGait applies single-path one-shot search to gait recognition in unconstrained environments [25],

while DCNAS searches cross-modality feature extraction and fusion structures for guided image super-resolution [26]. These approaches establish topology and operation choice as NAS design variables. GeoBridge investigates their combination within decoder stages whose external multi-scale connections remain fixed across the search space.

B. Architecture Transfer

Architecture transfer reuses design knowledge across datasets or prediction tasks. NAT adapts a pretrained supernet to target datasets [27], while NASNet transfers classification-oriented cells to object-detection backbones [15]. FNA++ combines parameter remapping with target-side architecture adaptation for dense prediction [17]. These methods illustrate different transfer units, from selected network configurations to reusable cells and adapted networks.

Cross-task studies examine which architectural choices remain useful when the prediction objective changes. TransNAS-Bench-101 evaluates architectures across multiple visual tasks [28], and PTF-EvoMDE transfers an encoder searched for depth estimation to classification, detection, and segmentation [22]. LAPT transfers architectural design principles [29]. GeoBridge studies the stage-internal topology of a decoder as the transfer unit between dense depth prediction and Gaussian reconstruction. Each target system retains the selected composition and primitive configurations and trains the instantiated network for its own objective.

C. Structural Encoding in EvoNAS

Architecture encoding determines the units on which evolutionary variation operates [30]. Direct encodings specify operations and connections explicitly. EvoCNN, for example, uses variable-length chromosomes to represent layer sequences [31]. NASNet searches reusable normal and reduction cells and stacks them into a full network [15]. Mainstream gradient-based NAS methods, including DARTS and P-DARTS, search over this cell-based encoding [7], [32]. Such encodings expose local architectural choices to crossover and mutation, while the decoding procedure determines how those choices form a valid network.

Early work on evolving artificial neural networks also used indirect encodings, in which the genome describes a construction process instead of listing every unit and connection. Kitano encodes a connectivity matrix through graph-rewriting rules [33], and cellular encoding grows a network from a single cell by a program of division and modification instructions [34]. NEAT instead evolves explicit node and connection genes, with historical markings that align genomes during crossover [35], and HyperNEAT generates connection weights from a function of neuron coordinates [36].

More recent encodings expose larger structural units through composition. SLGE constructs architectures with grammar-based transformations [37], hierarchical NAS assembles motifs from lower-level operations [9], and tree-based genetic programming has been applied to encoder–decoder networks [38]. GeoBridge adopts function composition within a fixed multi-scale decoder scaffold. Its residual, sequential, and parallel

nodes preserve the stage interface, so both a complete tree and any subtree define executable feature transformations. This organization supports structural variation at several levels without changing the connections between decoder stages.

D. Efficient EvoNAS

Candidate evaluation is a major cost in population-based NAS. Multi-fidelity evaluation reduces the resources allocated to candidate training [39]. Weight sharing reduces this cost by evaluating candidates within a trained supernet. SPOS uses uniformly sampled single-path subnetworks [40], while Once-for-All supports elastic depth, width, kernel size, and input resolution [41]. BossNAS divides the supernet into blocks trained by self-supervised ensemble bootstrapping, which improves architecture ranking in hybrid convolution–transformer spaces [42]. WIGEvoNAS uses weight inheritance to shorten candidate training in each generation of evolutionary search [43]. Applying these strategies to composed decoder trees requires a supernet that specifies how parameters are shared across different structural positions. GeoBridge instead evaluates independently instantiated candidates and uses a training-free proxy to control their evaluation cost.

Surrogate-assisted EvoNAS predicts candidate quality from previously evaluated architectures. Sun et al. [44] use an offline random-forest predictor, and Ma et al. [45] learn Pareto-wise dominance relationships for multi-objective search. CFMOGP-NAS combines the forecasts of multiple regression models within multi-objective genetic programming [46], and Li et al. [47] replace performance regression with a classifier when evolving fuzzy neural networks for defect recognition. These approaches require performance observations to train the estimator. Parameter-transfer methods such as PTF-EvoMDE reduce candidate-specific pretraining through hierarchical parameter mapping [22]. Training-free proxies offer a complementary route through responses measured at initialization, including connection sensitivity, gradient flow, activation statistics, and combinations of such criteria [48]–[52]. GSAP combines a parameter-scaled sensitivity statistic with gradient–parameter alignment under the regression stimulus used in the source search.

III. PROPOSED METHOD

GeoBridge evolves the internal topology of a decoder while preserving the stage interfaces through which the decoder connects to a geometric prediction system. Fig. 1 summarizes the source search and target reuse. We first define the source search and its multi-scale scaffold, then describe the composition space and training-free evaluator. The evolutionary procedure selects structures from this space, and target-specific training tests their reuse in depth estimation and Gaussian reconstruction.

A. Design Object and Source Scaffold

The source search jointly varies stage topology and primitive configuration under a fixed decoder scaffold. Let $\mathcal{D}_{\text{src}}$ denote the source dataset and Ω_α the composition space

defined in Section III-B. An architecture $\alpha \in \Omega_\alpha$ specifies both the composition of each stage and the configuration of its primitive transformations. With $\mathbf{w}^*$ denoting parameters optimized on the source task, the training-based reference problem is

$$\begin{aligned} & \underset{\alpha \in \Omega_\alpha}{\text{minimize}} && \{f_1(\alpha; \mathbf{w}^*), C_{\text{MAC}}(\alpha), C_{\text{Param}}(\alpha)\}, \\ & \text{subject to} && \mathbf{w}^* \in \arg \min_{\mathbf{w} \in \Omega_w} \mathcal{L}(\mathbf{w}; \alpha). \end{aligned} \quad (1)$$

Here, $f_1(\alpha; \mathbf{w}^*)$ measures depth prediction error after training, while $C_{\text{MAC}}(\alpha)$ and $C_{\text{Param}}(\alpha)$ measure decoder computation and parameter cost. These objectives describe the trade-off between predictive quality and resource use. GeoBridge maintains a non-dominated set, denoted by $\mathcal{Q}_{\text{src}}$, from which decoder structures with different cost profiles are selected.

The selected architecture is the object reused across target systems. For $\alpha^* \in \mathcal{Q}_{\text{src}}$, the stage compositions and primitive configurations remain fixed, while each target system optimizes its network weights and prediction interface. Thus, source search chooses the computation structure, and target training determines its parameters for each prediction objective.

Direct evaluation of Eq. (1) requires training every candidate decoder. To reduce this cost, GeoBridge uses J_{GSAP} from Section III-C as the quality objective during evolution. The resulting search minimizes $\{J_{\text{GSAP}}, C_{\text{MAC}}, C_{\text{Param}}\}$ over independently instantiated candidates. Section IV-D evaluates the association between the proxy and retrained depth accuracy.

All candidate decoders share the multi-scale scaffold shown in Fig. 2. DINOv2 ViT-S [53] supplies intermediate features from blocks B2, B5, B8, and B11, which are projected to scales of 1/4, 1/8, 1/16, and 1/32. Four Tree-Mamba stages process these features along a coarse-to-fine pathway. TartanAir images and cached features from a frozen Depth Anything V2 teacher provide the source inputs and auxiliary regression targets used for candidate evaluation.

The computation inside each Tree-Mamba stage is the searchable part of the scaffold. A projection adapts the encoder feature, which is fused with the preceding stage’s upsampled output at all but the coarsest scale. The stage then transforms this fused feature through its internal composition, and the finest-scale output is mapped to a prediction by the source head. The projections, cross-scale connections, upsampling operators, and head remain fixed during search. Consequently, candidates differ in how each stage processes its inputs while sharing the same information flow between stages.

B. Compositional Decoder Topologies

The topology space exposes the organization of transformations within each decoder stage. An architecture is represented by four functional trees, $\alpha = \{T_{32}, T_{16}, T_8, T_4\}$. Leaf nodes instantiate configurable Mamba primitives, and internal nodes define residual, sequential, or parallel composition. These choices vary processing depth, branching, and residual paths while preserving the stage interface. Fig. 3 illustrates the mapping from the tree representation to executable computation.

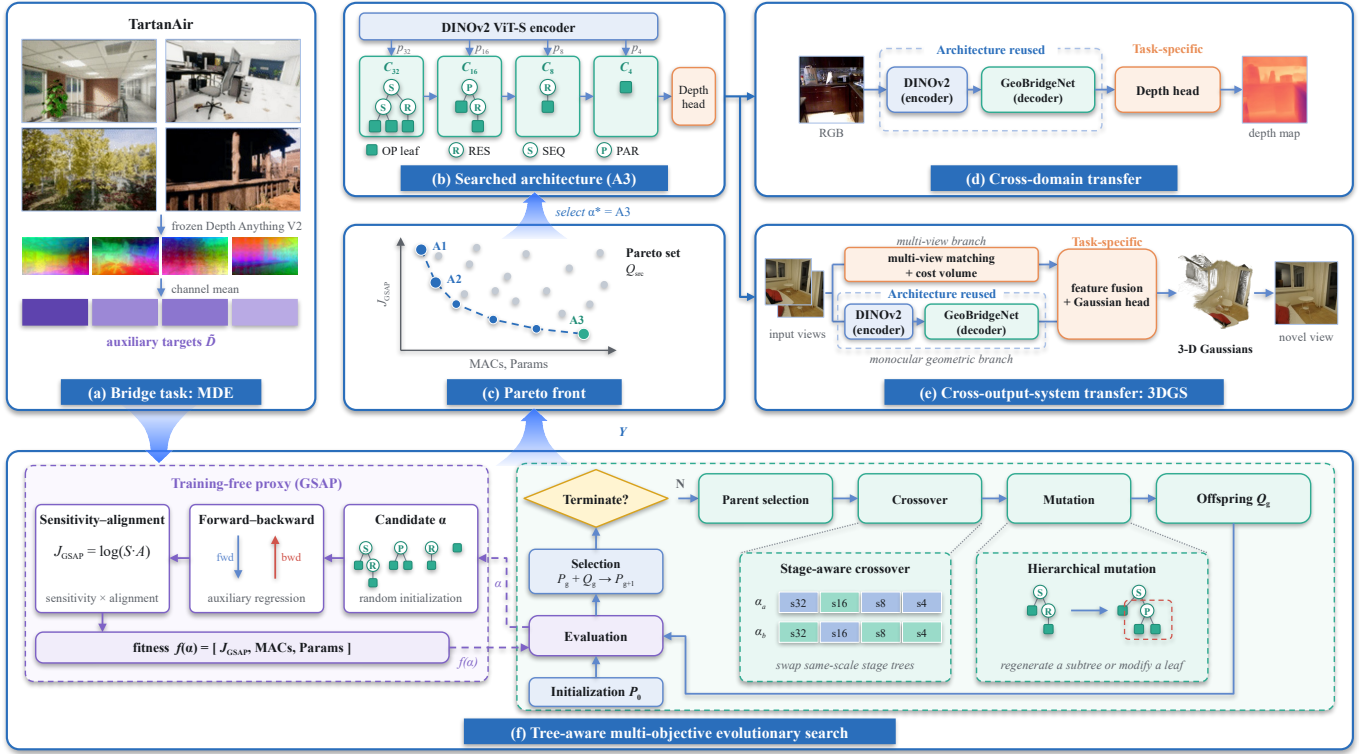


Fig. 1. GeoBridge searches once and instantiates the selected decoder topology in target systems. (a) TartanAir images and cached teacher-derived targets define the source evaluation environment. (f) NSGA-II varies stage compositions and primitive configurations through stage-aware crossover and hierarchical mutation; GSAP, MACs, and parameter count define candidate fitness. (c) The Pareto set supplies (b) the selected GeoBridgeNet-A3 structure. Its topology is retained for (d) depth estimation on real images and (e) the monocular branch of a Gaussian reconstruction system. Each target system trains its own weights and prediction interface.

1) *Stage-Wise Tree Structure:* The four functional trees are aligned with decoder resolutions of $1/32$, $1/16$, $1/8$, and $1/4$, denoted by T_{32} , T_{16} , T_8 , and T_4 , respectively. Each tree receives the fused feature of the corresponding Tree-Mamba stage and returns a representation with the same external interface for subsequent coarse-to-fine decoding. In this way, the fixed multi-scale scaffold determines how information flows across stages, whereas the functional tree determines how features are transformed within each stage.

Stage-specific bounds control the structural capacity of the decoder. The coarse stages T_{32} and T_{16} permit at most two levels of internal composition, while T_8 and T_4 permit at most one. With the child order of PAR treated as equivalent, the four one-level structures are OP, RES(OP), SEQ(OP,OP), and PAR(OP,OP). Each coarse stage therefore admits $1 + 4 + 4^2 + 4(4 + 1)/2 = 31$ structures, which gives $31^2 \times 4^2 = 15,376$ four-stage topologies. Including the primitive domains in Table I gives approximately 2.5×10^{26} configurations. These bounds allocate more compositional capacity to coarse stages, where spatial processing is less expensive.

2) *Composition Nodes:* The internal nodes define how primitive or composed transformations are organized into higher-level functions. Let $\mathcal{F}$ and $\mathcal{G}$ denote two shape-preserving child transformations with input $\mathbf{x}$. The three composition operators are defined as

$$\text{RES}(\mathbf{x}) = \mathbf{x} + \mathcal{F}(\mathbf{x}), \quad (2)$$

$$\text{SEQ}(\mathbf{x}) = \mathcal{G}(\mathcal{F}(\mathbf{x})), \quad (3)$$

and

$$\text{PAR}(\mathbf{x}) = \frac{1}{2} [\mathcal{F}(\mathbf{x}) + \mathcal{G}(\mathbf{x})]. \quad (4)$$

The composition rules control distinct aspects of stage topology. RES adds an identity path around a transformation, SEQ places transformations along a longer processing path, and PAR aggregates two branches that receive the same input. Each composition returns a function with the same input and output shape as its children. This closure property makes a subtree an executable transformation and permits structural variation within the stage's depth bound without additional interface matching.

3) *Primitive Leaves:* The leaves provide the elementary transformations from which the functional hierarchy is constructed. Each OP leaf instantiates a configurable Mamba-style spatial operator based on selective state-space modeling [54]. Its computational template is fixed, while six discrete variables determine its behavior: the number of stacked blocks, the set of two-dimensional scan paths, the fusion strategy for path outputs, the sharing of path-wise input projections, the post-scan state-fusion strategy, and the kernel size of the depth-wise

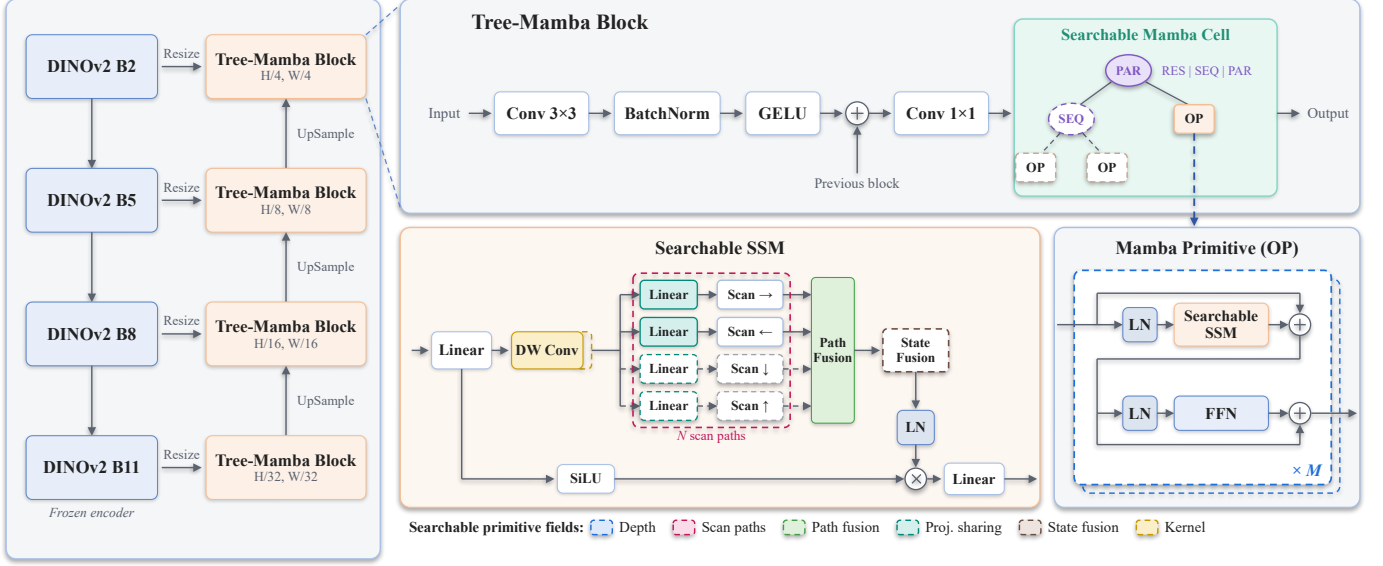


Fig. 2. Fixed scaffold and searchable stage computation in GeoBridge. DINOv2 ViT-S blocks B2, B5, B8, and B11 supply features resized to 1/4–1/32 of the input resolution. Lateral fusion and upsampling define the connections between four Tree-Mamba stages. Inside each stage, a functional tree determines the composition of Mamba primitives. Search varies this internal topology and its primitive configurations while retaining the scaffold. Selected structures are trained with the prediction interface required by each target system.

TABLE I
PRIMITIVE CONFIGURATION FIELDS OF THE FUNCTIONAL-TREE SEARCH SPACE. THE STAGE SCAFFOLD REMAINS FIXED WHILE EACH OP LEAF SELECTS A SCALE-VALID CONFIGURATION.

Field	Candidate Values	Scale Constraint
depth	{1, 2}	s_{32}/s_{16} : 1 or 2; s_8/s_4 : fixed to 1
d_{state}	{8, 16}	Independent per OP leaf
path set	fwd, fwd_bwd, fwd_bwd_hw	s_{32}/s_{16} : FB or FB-HW; s_8/s_4 : F or FB
fuse mode	scalar, input_gate	Independent per OP leaf
share mode	independent, share_x	Independent per OP leaf
state fusion	none, post	Independent per OP leaf
d_{conv}	{3, 5}	Independent per OP leaf

convolution. The corresponding candidate domains and scale-dependent constraints are summarized in Table I.

Primitive configuration and composition provide two levels of decoder design. The primitive fields determine the computation performed at each leaf, while the internal nodes determine how leaf and subtree outputs are combined. Coarse stages permit deeper primitives and more scan patterns, whereas fine stages use lighter configurations to limit high-resolution computation. Evolution therefore chooses both the elementary transformations and their organization within each stage.

C. GSAP for Efficient Candidate Evaluation

GSAP reduces the cost of evaluating independently instantiated decoder trees. Candidate-specific training would require repeated optimization for every combination of primitive configuration and functional composition. Instead, GSAP measures each candidate’s response to a fixed regression stimulus at initialization. The evaluator combines parameter-scaled gradient sensitivity with gradient–parameter alignment

and aggregates the responses from three independent initializations.

The auxiliary target provides the same regression stimulus to every candidate. Let $\mathcal{B} = \{\mathbf{x}_b\}_{b=1}^B$ be a fixed source batch from TartanAir. A frozen Depth Anything V2 model [55] extracts a feature representation $\mathbf{F}_b^T$ for each image. Spatial aggregation followed by channel averaging produces a scalar t_b , which is broadcast over the prediction resolution:

$$\mathbf{Y}_b^T = t_b \mathbf{1}, \quad (5)$$

where $\mathbf{1}$ denotes an all-one matrix with the same spatial dimensions as the candidate output.

Given an architecture α , the frozen DINOv2 encoder and randomly initialized decoder produce a dense prediction $\hat{\mathbf{Y}}_b^\alpha$. The auxiliary regression objective is defined as

$$\mathcal{L}_{\text{aux}}(\alpha) = \frac{1}{BHW} \sum_{b=1}^B \left\| \hat{\mathbf{Y}}_b^\alpha - \mathbf{Y}_b^T \right\|_2^2. \quad (6)$$

The images and teacher-derived targets are cached before evolution and reused for all candidates. The auxiliary loss measures each randomly initialized decoder’s response to this common stimulus. Its gradients provide the statistics used to rank architectures before training.

For the r -th independent initialization, $r \in \{1, 2, 3\}$, a single backward pass of $\mathcal{L}_{\text{aux}}$ yields

$$\mathbf{g}^{(r)} = \nabla_{\boldsymbol{\theta}^{(r)}} \mathcal{L}_{\text{aux}}, \quad (7)$$

where $\boldsymbol{\theta}^{(r)}$ denotes the trainable parameters of the independently initialized candidate decoder. We characterize each raw response from two complementary perspectives.

First, the parameter-scaled gradient sensitivity is

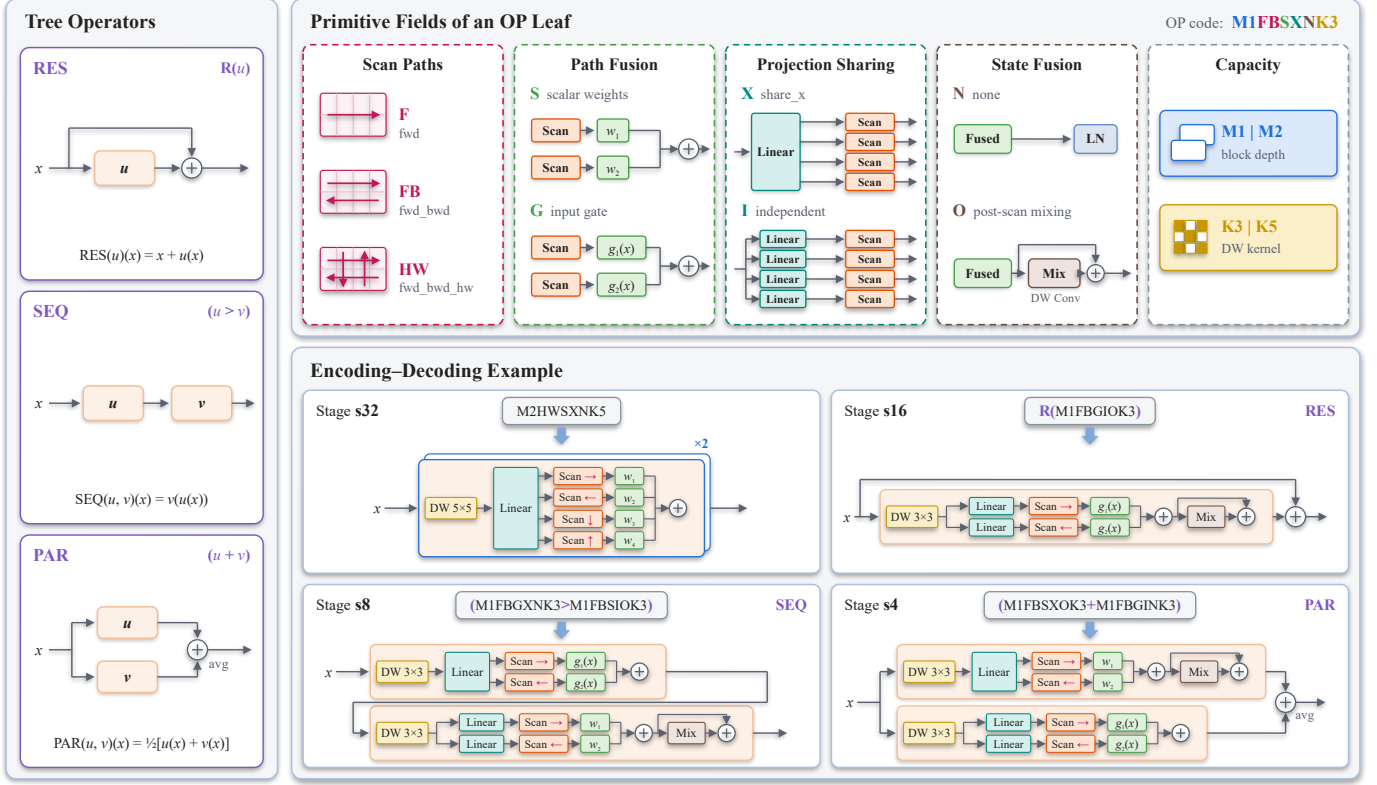


Fig. 3. Compositional decoder topology and its tree encoding. Left: RES, SEQ, and PAR define residual paths, serial processing, and parallel aggregation. Top right: the fields of an OP leaf specify its Mamba primitive. Bottom right: an example genotype is decoded into stage computation through recursive composition. Every subtree preserves its stage interface. Table I gives the primitive domains and scale-dependent bounds; Fig. 6 shows the evaluated structures.

$$S^{(r)}(\alpha) = \frac{1}{n_\alpha} \sum_{i=1}^{n_\alpha} |g_i^{(r)} \theta_i^{(r)}|, \quad (8)$$

where n_α is the number of trainable decoder parameters. The term $|g_i^{(r)} \theta_i^{(r)}|$ follows the parameter-scaled sensitivity used in connection-sensitivity criteria [48]. Averaging these terms gives a per-parameter statistic for each decoder.

Second, the directional coupling between the gradient and parameter vectors is

$$A^{(r)}(\alpha) = \frac{|\langle g^{(r)}, \theta^{(r)} \rangle|}{\|g^{(r)}\|_2 \|\theta^{(r)}\|_2}. \quad (9)$$

For each initialization, the untransformed GSAP response is $S^{(r)}(\alpha)A^{(r)}(\alpha)$. The three raw responses are first averaged, and the logarithm is applied only to their arithmetic mean:

$$J_{\text{GSAP}}(\alpha) = \log_{10} \left[\frac{1}{3} \sum_{r=1}^3 S^{(r)}(\alpha)A^{(r)}(\alpha) \right]. \quad (10)$$

The evolutionary search minimizes J_{GSAP} . An averaged raw response no greater than 10^{-12} is assigned a penalty value of 30. The relationship between the resulting rankings and retrained accuracy is evaluated in Section IV-D.

GSAP uses the same evaluation rule throughout evolution. The DINOv2 encoder and Depth Anything V2 teacher remain frozen, and only the decoder initialization changes between repeated measurements. Each measurement requires a forward

pass through the encoder and decoder, a backward pass through the decoder, and computation of the two gradient statistics. Three measurements yield one candidate score without updating the network parameters.

D. Evolving Decoder Topologies

Multi-objective evolution selects decoder compositions according to their GSAP response and resource costs. Each individual contains the four stage trees defined in Section III-B, and its objective vector is $\{J_{\text{GSAP}}, C_{\text{MAC}}, C_{\text{Param}}\}$. Algorithm 1 gives the population update, and Fig. 1 places this search within the complete source-to-target procedure.

The initial population is generated by recursively sampling valid functional trees under the stage-specific structural constraints defined in Section III-B. Internal nodes are selected from residual (RES), sequential (SEQ), and parallel (PAR) compositions, while operator (OP) nodes terminate the hierarchy with configurable Mamba primitives. The depth limits and primitive domains depend on spatial scale, which allows greater structural flexibility at coarse resolutions while constraining computation at finer resolutions. Each sampled stage tree preserves the interface of the decoder scaffold.

Each candidate is evaluated through three independent decoder initializations. Each initialization produces one raw GSAP response from a forward-backward pass on the common source batch. Their arithmetic mean is log-transformed to obtain J_{GSAP} , while MACs and parameter count are measured from the architecture. Section III-C defines this evaluation rule.

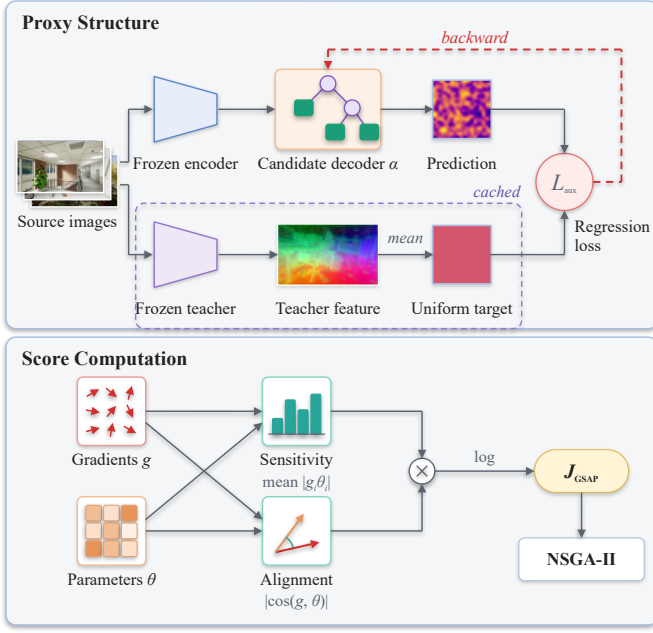


Fig. 4. Structure and computation of GSAP. Top: a fixed batch of source images passes through the frozen DINOv2 encoder and a randomly initialized candidate decoder, while the frozen Depth Anything V2 teacher provides a spatially uniform auxiliary target that is computed once and cached. One backward pass yields one raw sensitivity–alignment response without updating any parameter. Bottom: the illustrated sensitivity and alignment constitute one raw response; during search, this computation is repeated for three independent decoder initializations, the three raw products are averaged, and the logarithm is then applied to obtain J_{GSAP} , which forms the fitness together with MACs and parameters.

Evolutionary variation acts on the functional units defined by the trees. Binary tournament selection chooses parents using non-dominated rank and crowding distance. Stage-aware crossover exchanges an entire same-scale tree between two parents with probability p_c . Hierarchical mutation acts with probability p_m and can regenerate a subtree, modify an internal composition, or resample a primitive configuration. These operations vary stage computation while preserving the external interfaces of the decoder scaffold.

NSGA-II environmental selection maintains a population of proxy–cost trade-offs. The parent and offspring populations are merged, ranked by non-dominated sorting, and truncated using crowding distance. Thus, previously selected structures compete with newly generated candidates at each generation. Algorithm 1 summarizes the population update.

The final non-dominated front provides structures for downstream evaluation. GeoBridgeNet-A1 represents a low-cost design, A2 an intermediate design, and A3 a higher-capacity design with richer compositions. All three belong to the final non-dominated set. Before this front is extracted, the surviving population of the last generation is scored once more under the same three-initialization rule, and these evaluations are included in the recorded search totals. Section IV-A describes their stage structures and the search dynamics.

Algorithm 1: Evolving decoder topologies

Input: Population size N , number of generations G , crossover probability p_c , mutation probability p_m .

Output: Pareto set $\mathcal{Q}_{\text{src}}$.

```

1  $g \leftarrow 0$ ;
2  $P_0 \leftarrow$  Randomly initialize  $N$  functional-tree
  genotypes;
3 while  $g < G$  do
4   Evaluate each individual in  $P_g$  with three
     independent initializations and average the raw
     GSAP responses;
5   Rank  $P_g$  by non-dominated sorting and crowding
     distance;
6    $Q_g \leftarrow \emptyset$ ;
7   while  $|Q_g| < N$  do
8      $\{p_1, p_2\} \leftarrow$  Binary tournament selection( $P_g$ );
9      $\{q_1, q_2\} \leftarrow$  Stage-aware crossover( $p_1, p_2, p_c$ );
10     $\{q_1, q_2\} \leftarrow$  Hierarchical mutation( $q_1, q_2, p_m$ );
11     $Q_g \leftarrow Q_g \cup \{q_1, q_2\}$ ;
12  end
13  Evaluate each individual in  $Q_g$  with three
     independent initializations and average the raw
     GSAP responses;
14   $P_{g+1} \leftarrow$  Environmental selection( $P_g \cup Q_g$ );
15   $g \leftarrow g + 1$ ;
16 end
17 Evaluate each individual in  $P_G$  with three independent
     initializations and average the raw GSAP responses;
18  $\mathcal{Q}_{\text{src}} \leftarrow$  First non-dominated front of the evaluated  $P_G$ ;
19 return  $\mathcal{Q}_{\text{src}}$ 

```

E. Instantiation in Target Systems

Target instantiation retains the selected stage computation while connecting it to the target system’s prediction interface. For depth estimation on NYU Depth V2 and KITTI, an adaptive depth-bin head maps decoder features to depth, and the network is trained with ground-truth supervision. The decoder weights are initialized randomly, so the reused design consists of its topology and primitive configurations.

Gaussian NVS places the selected decoder in the monocular geometric branch of a DepthSplat-based system [6]. Its output is fused with multi-view information for depth and Gaussian prediction. The monocular branch first undergoes target-side distillation from Depth Anything V2 pseudo-depth, followed by optimization through differentiable rendering. The selected topology remains fixed through both stages. Scene-level Gaussian refinement and COS3D semantic optimization subsequently evaluate the downstream utility of the reconstructed representation [13], [14].

IV. EXPERIMENTS

The experiments examine which decoder structures emerge from source search and whether they retain their utility across geometric prediction systems. After characterizing the selected topologies, we evaluate depth estimation on NYU

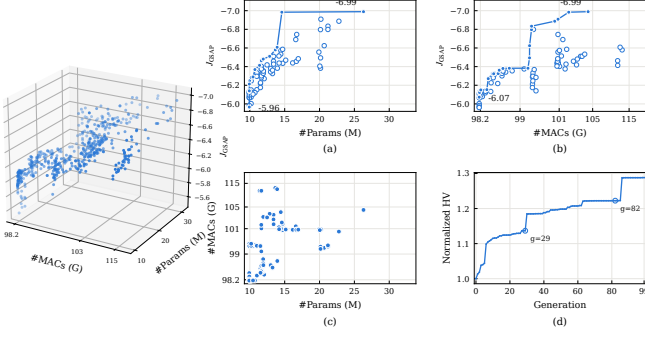


Fig. 5. Search dynamics of the 100-generation main run using three-repeat GSAP aggregation. Left: distribution of searched architectures in the three-objective space of J_{GSAP} , MACs, and parameters. (a)–(c) Pairwise projections of the same objective space, with the non-dominated boundary highlighted. (d) Normalized hypervolume over generations; generations 29 and 82 are marked to emphasize that improvement continues beyond the early search before the late-stage plateau. This run is separate from the encoding comparison in Table VIII.

Depth V2 [56] and KITTI [57] and Gaussian NVS on RealEstate10K [58]. Controlled studies compare decoder designs, evolutionary representations, and candidate evaluation. The final study evaluates the semantic utility of the reconstructed Gaussian scenes on LERF. Benchmark definitions, baselines, evaluation metrics, and training settings are provided in the supplementary material.

A. Topologies Selected by Source Evolution

The source experiment characterizes the structures and proxy–cost trade-offs produced by one architecture search. GeoBridge evolves a population of 64 decoders for 100 generations on the TartanAir source batch. The selected structures from this run are reused in all subsequent target systems.

The final population contains 26 non-dominated solutions. Fig. 5 shows that normalized HV increases early in the run and continues to improve after generations 29 and 82 before approaching a plateau. The objective-space projections show distinct trade-offs among J_{GSAP} , MACs, and parameter count. A1, A2, and A3 belong to this final non-dominated set.

The selected architectures instantiate different compositions within the common scaffold. As shown in Fig. 6, A1 and A2 mainly use individual primitives. A3 combines three primitives sequentially with residual refinement at s_{32} and uses parallel composition at s_{16} . Its finer stages contain a residual transformation at s_8 and one primitive at s_4 . These examples illustrate the design freedom introduced by stage-internal topology: a selected stage can retain a single transformation or compose several transformations along serial and parallel paths.

B. Reuse Across Geometric Prediction Systems

The reuse experiments retain the selected stage topologies and primitive configurations through the target training described in Section III-E. Depth estimation tests changes in image domain and depth range, while Gaussian NVS tests integration into a different prediction system. All three GeoBridgeNet variants are evaluated under both settings.

1) *Depth estimation across domains:* Cross-domain depth estimation tests decoder reuse under changes in scene statistics and depth range. The source search uses synthetic TartanAir images, whereas NYU Depth V2 contains real indoor scenes and KITTI contains real outdoor scenes. The target networks retain the searched decoder structures and use an adaptive depth-bin prediction head.

The controlled comparisons isolate decoder performance under a common encoder and training protocol. As reported in Tables II and III, A3 achieves the best AbsRel, RMSE, and δ_1 among the four DINOv2-adapted baselines. On NYU Depth V2, its AbsRel is 0.0876, compared with 0.0896 for DCDepth and 0.0902 for Metric3D v2. On KITTI, the corresponding values are 0.0555, 0.0585, and 0.0575. Metric3D v2 has lower latency, while A3 provides higher depth accuracy on both datasets.

Published systems place these results within a broader range of model sizes and training settings. On NYU Depth V2, A3 improves AbsRel over PixelFormer from 0.090 to 0.0876 with 42.5M parameters, while iDisc and IEBins with Swin-L reach 0.086 and 0.087. On KITTI, A3 reaches 0.0555, between AdaBins at 0.058 and the large Swin-based systems at 0.050–0.052. The NAS-based methods in the tables report higher AbsRel with smaller networks of 1.8–20.5M parameters. Their published search costs range from several GPU days for PTF-EvoMDE to more than 24 GPU days for the reproduced DARTS, DetNAS, and GDAS searches [7], [22], [59], [60]. GeoBridge uses approximately 12 GPU hours for its source search on an RTX 3090.

The three transferred structures retain different accuracy–cost profiles on both target datasets. From A1 to A3, NYU AbsRel decreases from 0.0918 to 0.0876 as latency increases from 36.9 to 40.7 ms. KITTI AbsRel decreases from 0.0582 to 0.0555 as latency increases from 53.9 to 59.1 ms. Thus, the source search supplies several useful decoder configurations for target deployment.

The qualitative comparisons illustrate boundary recovery in indoor and outdoor scenes. Figs. 7 and 8 compare A3 with IEBins, PixelFormer, and NeWCRFs. The highlighted NYU regions contain thin objects, cabinet edges, and a glass shower door, while the KITTI regions contain distant vehicles and roadside structures. A3 preserves visible object boundaries and local depth transitions in these examples.

2) *From depth decoding to Gaussian reconstruction:* Cross-system transfer reuses the decoder as a feature-processing component in Gaussian NVS. The same stage topology is instantiated in the monocular geometric branch of a DepthSplat-based pipeline. Its output is fused with multi-view information for depth and Gaussian prediction, and the full system is trained through differentiable rendering. This experiment evaluates the searched structure under a different output representation and training objective.

The transferred decoders achieve competitive reconstruction quality on RealEstate10K. Table IV reports PSNR values of 27.02, 27.07, and 27.11 dB for A1, A2, and A3, respectively. A3 exceeds pixelSplat by 1.22 dB and MVSplat by 0.72 dB. Among the DepthSplat variants [6], the ViT-S two-scale model reaches 27.12 dB with 41M parameters, and the ViT-L two-

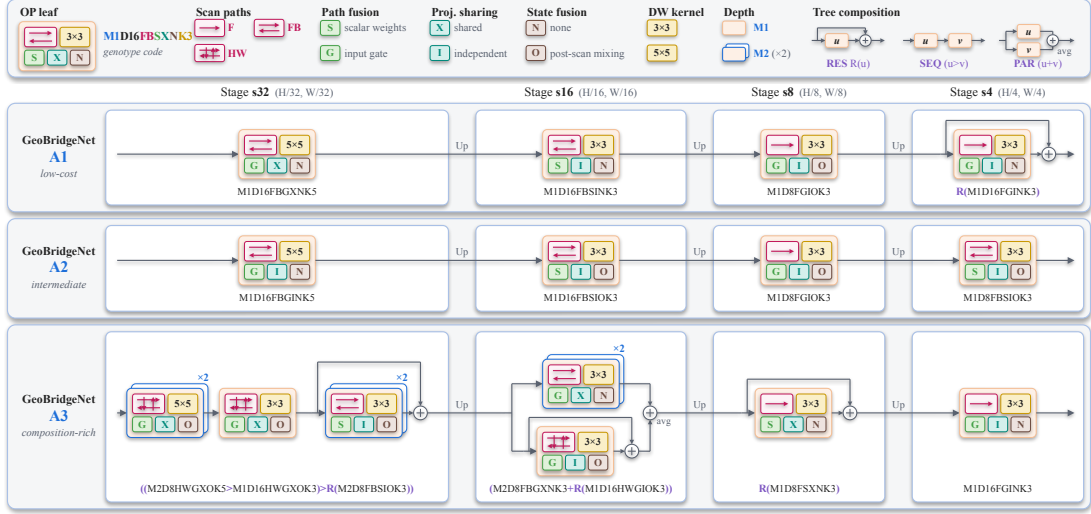


Fig. 6. Stage-wise architectures of GeoBridgeNet-A1, A2, and A3, decoded from their genotypes. Each row shows the searched Mamba cells of one variant in decoding order from s32 to s4; Up denotes the fixed lateral fusion and upsampling of the scaffold in Fig. 2. Each card is one OP leaf, whose scan paths, depth-wise kernel, path fusion, projection sharing, and state fusion follow the legend, and a stacked card denotes a two-block primitive. RES, SEQ, and PAR compositions are drawn as in Fig. 3, and the genotype string of each cell is given beneath it.

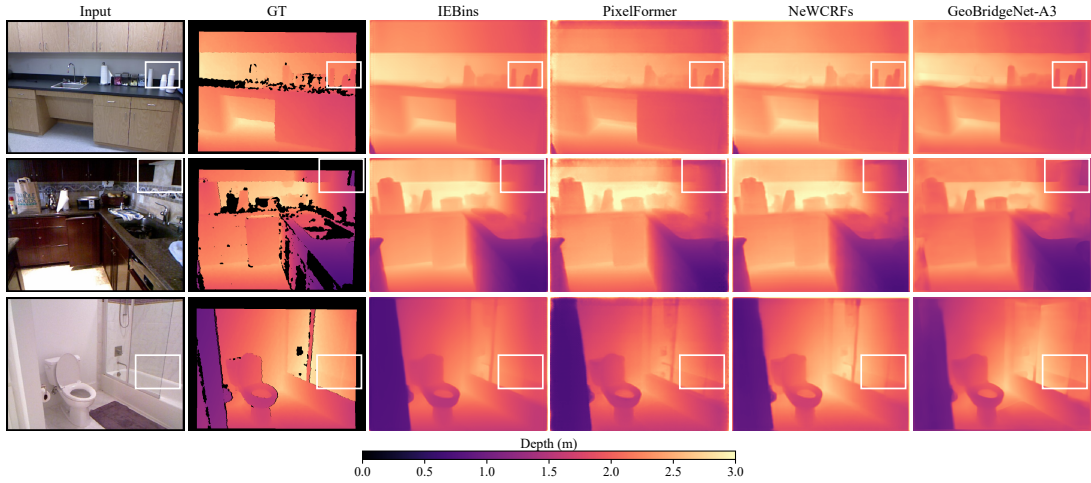


Fig. 7. NYU Depth V2 qualitative comparison. White boxes mark boundary differences within 0–3 m.

scale model reaches 27.47 dB with 354M parameters. The 62.0M-parameter A3 therefore reaches PSNR close to the ViT-S two-scale DepthSplat variant.

The measured runtimes show different reconstruction–latency trade-offs across systems. A1, A2, and A3 require 70.6, 71.4, and 78.8 ms, respectively, compared with 62 ms for MVsplat, 141 ms for the large DepthSplat variant, and 221.6 ms for IDEsplat. The GeoBridgeNet variants are non-dominated in PSNR and latency among the entries with measured runtime. Within the family, A3 has the highest PSNR, while A2 has the best SSIM and LPIPS.

A matched control isolates the decoder. Table V replaces the monocular decoder of the same DepthSplat-based pipeline with a DPT decoder [75]. The encoder weights, multi-view branch, fusion interface, and Gaussian head are unchanged, and each variant is trained three times under identical data, teacher, losses, and optimization steps. The DPT variant

reaches 27.03 ± 0.09 dB PSNR, 0.881 ± 0.003 SSIM, and 0.119 ± 0.003 LPIPS with 52.0M parameters. A3 reaches 27.11 ± 0.05 dB, 0.878 ± 0.002 , and 0.121 ± 0.002 with 62.0M parameters. A3 exceeds the DPT control in all three matched runs, with per-run PSNR margins of 0.13, 0.09, and 0.02 dB. A3 has a higher mean PSNR, while DPT has better mean SSIM and LPIPS and fewer parameters. The transferred topology therefore yields a different reconstruction–cost trade-off within the same pipeline.

The qualitative results show that the transferred decoder supports spatial detail recovery in the rendering-based pipeline. Fig. 9 compares A3 with pixelSplat and MVsplat on wall decoration, foliage behind a screen door, and bedding. A3 recovers visible texture and edge structure in these regions. The depth and NVS evaluations together demonstrate reuse of the selected computation under changes in both scene domain and prediction interface.

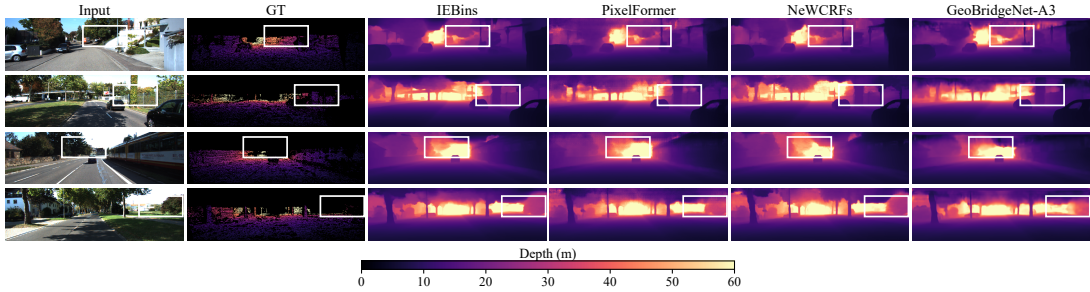


Fig. 8. KITTI qualitative comparison. White boxes mark boundary differences within 0–60 m.



Fig. 9. RealEstate10K novel-view comparison for GeoBridgeNet-A3. Red boxes mark geometric and detail differences.

C. Decoder Design and Evolutionary Representation

The controlled studies examine decoder design and the evolutionary representation used to explore it. Alternative decoder designs are first compared under a common depth-training protocol. Two search controls then separate the roles of topology freedom and evolutionary selection. A paired tree-flat study next evaluates how two encodings and their variation operators explore the same composition grammar, followed by downstream comparisons of their selected structures.

Comparison of Decoder Designs: The searched decoder achieves lower depth error than the alternative designs under the common NYU protocol. Table VI compares A3 with a searchable FPN, a fixed LocalMamba decoder, and adapted DPT and NeWCRFs decoders [3], [75]. Across three training seeds, A3 reaches an average AbsRel of 0.0876 ± 0.0002 , compared with 0.0929 ± 0.0013 , 0.0918 ± 0.0015 , 0.0981 ± 0.0011 , and 0.0912 ± 0.0020 , respectively. This comparison supports the effectiveness of the selected decoder within the shared

encoder and training setting.

Topology Freedom and Evolutionary Selection: Two controls separate the contribution of open topology from that of evolutionary search. Under the same encoder, primitive domain, GSAP evaluator, and 12,864 evaluation events, the first control evolves only the primitive configurations on a fixed stage template of residual and serial chains. The second samples structures independently from the full composition space with the initialization sampler of the main search. Each group repeats this search three times with independent random starts; every run selects one structure from its non-dominated set under a common limit on parameters and MACs by GSAP re-scoring, and every selected structure is trained under the NYU protocol of Section IV-B1. Target results do not enter the selection (Table VII).

Across the three searches, joint topology evolution obtains the best mean RMSE and δ_1 and matches the mean AbsRel of the fixed template. Compared with fixed-topology evolution, it

TABLE II

QUANTITATIVE COMPARISON ON NYU DEPTH V2. PARAMS DENOTES WHOLE-MODEL PARAMETERS (ENCODER, PROJECTIONS, DECODER, AND HEAD). REPORTED BASELINE LATENCIES USE THE SAME RTX 3090 BATCH-1 FP32 PROTOCOL. $\dagger$ DENOTES OUR DINOv2-ADAPTED REPRODUCTION OF THE CORRESPONDING PUBLISHED DECODER. VALUES FOR THE GEOBRIDGEnET ROWS ARE MEANS $\pm$ STANDARD DEVIATIONS OVER THREE TRAINING RUNS.

Method	AbsRel $\downarrow$	RMSE $\downarrow$	$\delta_1 \uparrow$	Params(M)	MACs(G)	Latency (ms)
BTS [61]	0.110	0.392	0.885	66.5	132.3	28.2
AdaBins [2]	0.103	0.364	0.903	78.3	187.8	49.2
DepthFormer [62]	0.096	0.339	0.921	274.1	1499.0	–
LocalBins [63]	0.099	0.357	0.907	73.6	186.2	32.9
P3Depth [64]	0.104	0.356	0.898	94.3	288.3	35.8
iDisc (Swin-T) [65]	0.109	0.377	0.894	39.0	218.8	53.4
iDisc (Swin-L) [65]	0.086	0.313	0.940	–	–	–
IEBins (Swin-T) [66]	0.108	0.375	0.893	90.7	378.2	52.0
IEBins (Swin-L) [66]	0.087	0.314	0.936	273.0	–	–
NeWCRFs [3]	0.095	0.334	0.922	270.4	281.1	54.5
BinsFormer [67]	0.094	0.330	0.925	254.6	290.7	79.0
PixelFormer [68]	0.090	0.322	0.929	270.9	273.5	42.8
CLIP2Depth [69]	0.100	0.379	0.900	–	–	–
LitFormer [70]	0.089	0.313	0.932	–	–	–
PTF-EvoMDENet [22]	0.113	0.406	0.876	6.1	–	–
Mde-EvoNAS [21]	0.142	0.471	0.824	20.5	–	–
LiDNAS [20]	0.132	0.487	0.845	2.1	–	–
DARTS [7]	0.128	0.449	0.844	5.5	–	–
DetNAS [59]	0.114	0.407	0.875	6.3	–	–
GDAS [60]	0.123	0.433	0.854	5.7	–	–
Depth Pro † [71]	0.0941	0.3331	0.9256	43.3	388.4	46.0
DCDepth † [72]	0.0896	0.3237	0.9337	87.5	414.7	235.4
Metric3D v2 † [73]	0.0902	0.3157	0.9330	33.6	94.0	24.3
InfiniDepth † [74]	0.1017	0.3628	0.9067	26.7	310.9	57.8
GeoBridgeNet-A1	0.0918 $\pm$ 0.0005	0.3252 $\pm$ 0.0016	0.9315 $\pm$ 0.0010	32.9	98.2	36.9
GeoBridgeNet-A2	0.0895 $\pm$ 0.0004	0.3207 $\pm$ 0.0011	0.9331 $\pm$ 0.0007	34.0	99.5	38.0
GeoBridgeNet-A3	0.0876 $\pm$ 0.0002	0.3134 $\pm$ 0.0005	0.9364 $\pm$ 0.0005	42.5	100.8	40.7

TABLE III

QUANTITATIVE COMPARISON ON THE KITTI EIGEN SPLIT UNDER THE 0–80 M PROTOCOL. PARAMS DENOTES WHOLE-MODEL PARAMETERS. BASELINE MACS AND LATENCIES FOLLOW THE MEASUREMENT PROTOCOL OF TABLE II. $\dagger$ DENOTES OUR DINOv2-ADAPTED REPRODUCTION OF THE CORRESPONDING PUBLISHED DECODER. VALUES FOR THE GEOBRIDGEnET ROWS ARE MEANS $\pm$ STANDARD DEVIATIONS OVER THREE TRAINING RUNS.

Method	AbsRel $\downarrow$	RMSE $\downarrow$	$\delta_1 \uparrow$	Params(M)	MACs(G)	Latency (ms)
BTS [61]	0.061	2.834	0.954	66.5	184.3	32.9
AdaBins [2]	0.058	2.360	0.964	78.3	261.7	47.2
DepthFormer [62]	0.052	2.143	0.975	274.1	–	–
P3Depth [64]	0.071	2.842	0.953	94.3	401.8	36.1
DPT [75]	0.062	2.573	0.959	123.1	319.3	60.1
iDisc (Swin-L) [65]	0.050	2.067	0.977	–	–	–
IEBins (Swin-L) [66]	0.050	2.011	0.978	273.0	–	–
NeWCRFs [3]	0.052	2.129	0.974	270.4	395.9	74.8
BinsFormer [67]	0.052	2.098	0.974	254.6	409.4	97.5
PixelFormer [68]	0.051	2.081	0.976	270.9	385.4	60.9
CLIP2Depth [69]	0.074	2.948	0.938	–	–	–
PTF-EvoMDENet [22]	0.061	2.470	0.957	6.3	–	–
Mde-EvoNAS [21]	0.069	2.823	0.944	17.6	–	–
LiDNAS [20]	0.133	5.157	0.842	1.8	–	–
DARTS [7]	0.069	2.763	0.945	6.2	–	–
DetNAS [59]	0.063	2.579	0.954	6.0	–	–
GDAS [60]	0.066	2.651	0.951	5.8	–	–
Depth Pro † [71]	0.0794	2.6711	0.9574	43.3	555.4	67.0
DCDepth † [72]	0.0585	2.2116	0.9710	87.5	592.1	299.2
Metric3D v2 † [73]	0.0575	2.2405	0.9714	33.6	145.3	36.6
InfiniDepth † [74]	0.0623	2.4290	0.9596	26.7	447.4	83.8
GeoBridgeNet-A1	0.0582 $\pm$ 0.0005	2.2411 $\pm$ 0.0096	0.9694 $\pm$ 0.0008	32.9	130.5	53.9
GeoBridgeNet-A2	0.0561 $\pm$ 0.0004	2.2153 $\pm$ 0.0085	0.9702 $\pm$ 0.0007	34.0	134.2	55.0
GeoBridgeNet-A3	0.0555 $\pm$ 0.0003	2.1993 $\pm$ 0.0082	0.9719 $\pm$ 0.0006	42.5	140.5	59.1

lowers RMSE by 0.0036 and raises δ_1 by 0.0027 at comparable MACs, with 15% more parameters. Compared with random sampling, which evaluates 2.2 times as many distinct decoder configurations, it lowers AbsRel by 0.0009 and raises δ_1 by 0.0032 with 16% fewer parameters. These mean results favor joint evolution on RMSE and δ_1 at similar MACs, with parameter costs between those of the two controls.

Structural Exploration: The tree–flat comparison holds the available topologies fixed and varies their encoding and evolutionary operators. We conduct 15 paired runs with the same Mamba grammar, NSGA-II configuration, evaluation budget, and proxy–cost objectives. The flat encoding uses fixed heap slots with one-point crossover and locus mutation. The tree encoding uses stage-aware crossover and hierarchical mutation.

Tree-aware evolution visits more distinct genotypes and

TABLE IV

SYSTEM-LEVEL NOVEL-VIEW SYNTHESIS COMPARISON ON REALSTATE10K. PARAMS DENOTES THE COMPLETE FEED-FORWARD PIPELINE. DEPTHSPAT (ViT-S) AND DEPTHSPAT (ViT-L) DENOTE THE PUBLISHED ViT-S AND ViT-L TWO-SCALE VARIANTS. A DASH INDICATES AN UNMEASURED LATENCY. VALUES FOR THE GEOBRIDGEnET ROWS ARE MEANS $\pm$ STANDARD DEVIATIONS OVER THREE TRAINING RUNS.

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Params	Latency (ms)
pixelNeRF [76]	20.43	0.589	0.550	–	–
GPNR [77]	24.11	0.793	0.255	–	–
AttnRend [78]	24.78	0.820	0.213	–	–
MuRF [79]	26.10	0.858	0.143	–	–
pixelSplat [4]	25.89	0.858	0.142	119M	162
MVSplat [5]	26.39	0.869	0.128	12M	62
DepthSplat (ViT-S) [6]	27.12	0.884	0.119	41M	–
DepthSplat (ViT-L) [6]	27.47	0.889	0.114	354M	141
IDESplat [80]	27.80	0.893	0.108	37.6M	221.6
GeoBridgeNet-A1	27.02 $\pm$ 0.06	0.879 $\pm$ 0.002	0.119 $\pm$ 0.003	52.4M	70.6
GeoBridgeNet-A2	27.07 $\pm$ 0.04	0.881 $\pm$ 0.002	0.118 $\pm$ 0.002	53.5M	71.4
GeoBridgeNet-A3	27.11 $\pm$ 0.05	0.878 $\pm$ 0.002	0.121 $\pm$ 0.002	62.0M	78.8

TABLE V

MATCHED DECODER CONTROL FOR GAUSSIAN NOVEL-VIEW SYNTHESIS ON REALSTATE10K. BOTH ROWS USE THE SAME DEPTHSPAT-BASED PIPELINE, DATA, TEACHER, LOSSES, AND 300K OPTIMIZATION STEPS. PARAMS DENOTES THE COMPLETE FEED-FORWARD PIPELINE. VALUES ARE MEANS $\pm$ STANDARD DEVIATIONS OVER THREE PAIRED TRAINING RUNS.

Monocular decoder	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Params
DPT [75]	27.03 $\pm$ 0.09	0.881 $\pm$ 0.003	0.119 $\pm$ 0.003	52.0M
GeoBridgeNet-A3	27.11 $\pm$ 0.05	0.878 $\pm$ 0.002	0.121 $\pm$ 0.002	62.0M

topologies under the fixed evaluation budget. Table VIII reports 839 ± 17 genotypes and 276 ± 21 four-stage topologies per run, compared with 455 ± 33 and 147 ± 32 for the flat encoding, and a topology entropy of 4.67 compared with 4.17. The paired tests detect significant differences in all three coverage measures. Best GSAP and HV do not differ significantly in these paired tests ($p = 0.68$ and $p = 0.98$). The detected advantage is broader structural coverage.

Reuse of Selected Tree and Flat Structures: The selected tree and flat structures exhibit different depth-estimation profiles after transfer. Under the same downstream protocols, the tree-derived A3 obtains lower RMSE and higher δ_1 on NYU Depth V2, while the flat structure obtains lower AbsRel (Table IX). On KITTI, A3 obtains better values on all three reported metrics.

Cross-system results also show metric-dependent differences between the selected structures. On RealEstate10K, the tree-derived structure improves PSNR by 0.18 dB, while the flat structure has slightly better SSIM and LPIPS (Table X). After LERF refinement, the tree initializer has 2.58 dB higher average PSNR. Its semantic mIoU is 0.80 percentage points higher, while mAcc is 4.35 points lower. These comparisons identify reconstruction and semantic trade-offs between the two selected structures.

D. GSAP as a Search Signal

The proxy study examines the association between GSAP and retrained depth accuracy, its variation across evaluation conditions, and its computational cost. A separate benchmark study evaluates its behavior across visual tasks.

Ranking Reliability: GSAP provides the strongest ranking association among the evaluated proxies on the retrained

TABLE VI

DECODER COMPARISON ON NYU DEPTH V2. ALL DECODERS USE DINOv2 ViT-S, THE SAME ENCODER ADAPTATION, AND THE SAME RETRAINING PROTOCOL; ABSREL IS REPORTED AS THE MEAN AND STANDARD DEVIATION OVER THREE SEEDS, TOGETHER WITH THE BEST SEED.

Decoder Design	AbsRel (Mean $\pm$ Std.) $\downarrow$	Best Seed $\downarrow$
Conventional FPN blocks	0.0929 $\pm$ 0.0013	0.0917
Fixed LocalMamba decoder	0.0918 $\pm$ 0.0015	0.0904
DPT decoder adaptation	0.0981 $\pm$ 0.0011	0.0968
NeWCRFs decoder adaptation	0.0912 $\pm$ 0.0020	0.0894
Searched Mamba tree (GeoBridgeNet-A3, ours)	0.0876 $\pm$ 0.0002	0.0874

TABLE VII

CONTROLLED SEARCH COMPARISON ON NYU DEPTH V2. DISTINCT IS THE NUMBER OF DISTINCT STRUCTURES EVALUATED; COST IS THE GSAP EVALUATION TIME ON ONE H20 GPU. VALUES ARE MEANS $\pm$ STANDARD DEVIATIONS OVER THREE SEARCHES PER GROUP.

Search	Distinct	Cost (h)	Params/MACs	AbsRel $\downarrow$	RMSE $\downarrow$	$\delta_1 \uparrow$
Joint evolution (ours)	5,729$\pm$83	7.7$\pm$0.4	14.5M/99.0G	0.0880 $\pm$ 0.0048	0.3179 $\pm$ 0.0079	0.9382 $\pm$ 0.0041
Fixed topology	4,669 $\pm$ 64	6.8 $\pm$ 0.3	12.6M/99.2G	0.0880 $\pm$ 0.0042	0.3215 $\pm$ 0.0071	0.9355 $\pm$ 0.0038
Random search	12,864	7.7 $\pm$ 0.5	17.2M/98.9G	0.0889 $\pm$ 0.0051	0.3180 $\pm$ 0.0083	0.9350 $\pm$ 0.0044

decoder sample. We sample 48 architectures independently of the evolutionary search and retrain all of them for 14,150 steps. Over 24 initialization seeds, Table XI reports a mean Spearman correlation of 0.344 between $-J_{\text{GSAP}}$ and retrained δ_1 , with 67% of the draws significant at $p < 0.05$. GSAP also obtains the highest Hit@10% and NDCG@10%. The remaining proxies have mean correlations between -0.011 and 0.184 , and at most 13% of their draws are significant.

Stability and Efficiency: The evaluated rankings remain consistent across the tested source batches. With three initializations per candidate, GSAP has a mean pairwise rank correlation of 0.936 across four batches, with a minimum of 0.893. These measurements characterize variation under the tested initialization and batch conditions.

GSAP reduces evaluation cost to three forward-backward measurements per candidate. Each measurement requires approximately 1.1 s on an RTX 3090, so the aggregated candidate score requires about 3.3 s. The complete 100-generation source search takes approximately 12 GPU hours. A 1,000-step short-training baseline requires iterative optimization and obtains a Spearman correlation of -0.068 with the retrained accuracy.

Cross-Task Behavior: The TransNAS-Bench-101 study shows that proxy quality varies across prediction tasks. Table XII reports correlations over 7,352 benchmark architectures [28]. GSAP has the highest correlation among the compared proxies for autoencoding and room-layout prediction. Its correlations with object classification and semantic segmentation are positive, while GradSign obtains the highest values on those tasks. The strongest GSAP result in this study is on autoencoding.

E. Downstream Utility of the Transferred 3-D Representation

The LERF experiment evaluates the downstream utility of Gaussian representations produced by the transferred pipeline. Feed-forward predictions initialize scene-level 3DGS refinement, after which COS3D performs semantic feature optimization and open-vocabulary segmentation. This evaluation

TABLE VIII

COMPARISON OF TREE-AWARE EVOLUTION AND A FIXED-LENGTH ENCODING OVER 15 PAIRED RUNS, WITH 1,920 EVALUATIONS PER RUN. VALUES ARE MEANS $\pm$ STANDARD DEVIATIONS; p -VALUES ARE FROM PAIRED WILCOXON SIGNED-RANK TESTS. BOLD MARKS SIGNIFICANT DIFFERENCES. GENOTYPES COUNTS UNIQUE SERIALIZED GENOTYPES AND TOPOLOGIES COUNTS UNIQUE FOUR-STAGE TOPOLOGY TUPLES; ENTROPY IS THE SHANNON ENTROPY OF THE TOPOLOGY DISTRIBUTION. THIS COMPARISON IS SEPARATE FROM THE MAIN SEARCH IN FIG. 5.

Encoding	Best $J_{\text{GSAP}} \downarrow$	HV $\uparrow$	Genotypes	Topologies	Entropy
Tree-coded (ours)	-6.84 $\pm$ 0.20	0.80 $\pm$ 0.17	839 $\pm$ 17	276 $\pm$ 21	4.67 $\pm$ 0.17
Fixed-length vector	-6.79 $\pm$ 0.26	0.81 $\pm$ 0.23	455 $\pm$ 33	147 $\pm$ 32	4.17 $\pm$ 0.44
p (signed-rank)	0.68	0.98	$< 10^{-4}$	$< 10^{-3}$	0.002

TABLE IX

DOWNSTREAM TRANSFER OF STRUCTURES DISCOVERED BY TREE AND FLAT ENCODINGS UNDER THE SAME SEARCH, RETRAINING, AND EVALUATION SETTINGS ON NYU DEPTH V2 AND KITTI. VALUES ARE MEANS $\pm$ STANDARD DEVIATIONS OVER THREE TRAINING RUNS.

Encoding	NYU Depth V2			KITTI		
	AbsRel $\downarrow$	RMSE $\downarrow$	$\delta_1 \uparrow$	AbsRel $\downarrow$	RMSE $\downarrow$	$\delta_1 \uparrow$
Tree (GeoBridgeNet-A3)	0.0876 $\pm$ 0.0002	0.3134 $\pm$ 0.0005	0.9364 $\pm$ 0.0005	0.0555 $\pm$ 0.0003	2.1993 $\pm$ 0.0082	0.9719 $\pm$ 0.0006
Flat encoding	0.0870 $\pm$ 0.0003	0.3193 $\pm$ 0.0007	0.9348 $\pm$ 0.0006	0.0557 $\pm$ 0.0004	2.2185 $\pm$ 0.0104	0.9716 $\pm$ 0.0007

connects the reconstructed geometry to a subsequent semantic processing stage.

Scene-level refinement improves the photometric quality of the feed-forward initialization. As shown in Table XIII, the average PSNR of the A3-initialized representation rises from 14.90 to 23.33 dB after 30,000 iterations. Vanilla 3DGS reaches 24.40 dB under the same refinement budget. A3 reaches 24.29 dB on Figurines and 24.69 dB on Ramen, compared with 24.10 and 24.75 dB for vanilla 3DGS; the larger gaps occur on Teatime and Waldo Kitchen.

The refined representations support open-vocabulary segmentation in the COS3D pipeline. Table XIV reports 50.54% mIoU and 71.32% mAcc for the A3 initializer, compared with 50.76% and 72.08% for COS3D with kernel-regression inference. Among the transferred structures, A1 obtains the highest mIoU and Acc@0.5 at 51.06% and 53.67%, while A3 obtains the highest mAcc (Table XV). These results show that the reconstructed representations remain usable for downstream semantic processing.

The visual examples show object-level segmentation from the A3-initialized representation. Fig. 10 presents six open-vocabulary queries. The sheep, yellow pouf, stuffed bear, pot, and frog cup have IoUs between 86.3 and 97.6%. The dark cup reaches 74.4%, with scattered Gaussians around the rim and handle. These examples illustrate the outputs of the combined reconstruction and semantic pipeline.

V. CONCLUSION

This paper presented GeoBridge for evolving transferable decoder topologies in geometric vision. Residual, sequential, and parallel composition make the internal organization of each stage searchable while preserving the interfaces of a multi-scale scaffold. A functional-tree representation and GSAP support multi-objective evolution over these compositions at low evaluation cost. The selected structures retain their utility after target-specific training for real-world depth

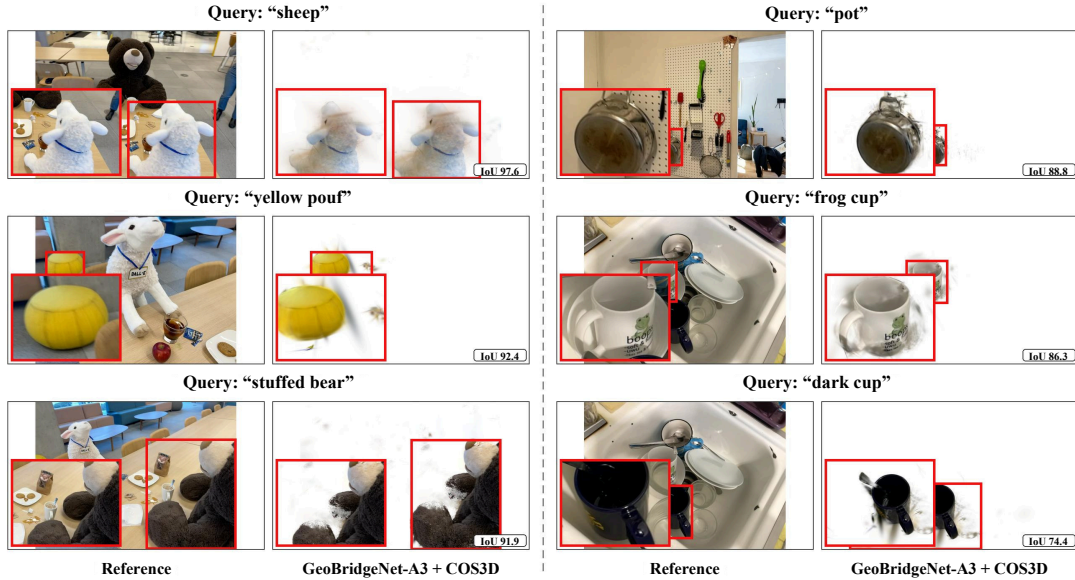


Fig. 10. Open-vocabulary 3-D segmentation and object isolation with the GeoBridgeNet-A3-initialized Gaussian model. Red boxes mark queried objects, and labels report IoU (%).

TABLE X

ENCODING-LEVEL TRANSFER COMPARISON ON REALESTATE10K. BOTH STRUCTURES ARE RETRAINED AND EVALUATED UNDER THE SAME NOVEL-VIEW SYNTHESIS PROTOCOL. VALUES ARE MEANS $\pm$ STANDARD DEVIATIONS OVER THREE TRAINING RUNS.

Encoding	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Tree (GeoBridgeNet-A3)	27.11 $\pm$ 0.05	0.878 $\pm$ 0.002	0.121 $\pm$ 0.002
Flat encoding	26.93 $\pm$ 0.06	0.879 $\pm$ 0.003	0.120 $\pm$ 0.003

TABLE XI

PROXY RANKINGS ON 48 RETRAINED NYU DEPTH V2 ARCHITECTURES. EACH COMPARED PROXY USES ONE INITIALIZATION PER SEED AND GSAP AVERAGES THREE; ρ IS THE MEAN $\pm$ STD OVER 24 SEEDS, AND $p < 0.05$ IS THE SHARE OF SEEDS REACHING SIGNIFICANCE.

Proxy	Spearman ρ	$p < 0.05$	Hit@10%	NDCG@10%
GSAP (ours)	0.344$\pm$0.143	67%	0.27	0.701
NASWOT [50]	0.161 $\pm$ 0.070	0%	0.08	0.572
SNIP [48]	0.054 $\pm$ 0.143	4%	0.12	0.547
GraSP [49]	0.184 $\pm$ 0.125	13%	0.15	0.625
AZ-NAS (aggregated) [52]	0.043 $\pm$ 0.122	0%	0.19	0.652
GradNorm (large-gradient) [51]	0.070 $\pm$ 0.146	0%	0.12	0.553
SynFlow [81]	-0.011 $\pm$ 0.143	4%	0.10	0.487

estimation and Gaussian novel-view synthesis, and their reconstructed scenes support downstream semantic processing. These results support stage-internal topology as a reusable unit of decoder design. Future work will examine which composition patterns account for transfer performance and how the source evaluation setting affects their selection.

TABLE XII

CROSS-TASK SPEARMAN CORRELATIONS OF PROXIES ON 7,352 TRANSNAS-BENCH-101 ARCHITECTURES.

Proxy	Autoencoder	Room layout	Object class.	Semantic seg.
SNIP [48]	0.0822	0.0215	0.3214	0.0122
GraSP [49]	0.0934	0.0287	0.2891	0.0256
GradNorm [51]	0.0911	0.0410	0.2647	0.0345
SynFlow [81]	0.1245	0.0532	0.3541	0.0412
NASWOT [50]	0.1534	0.1145	0.3812	0.0912
JacCov [50]	0.1876	0.1312	0.4125	0.1034
Zen-NAS [82]	0.1312	0.0712	0.3345	0.0623
GradSign [83]	0.3012	0.1723	0.4312	0.1134
GSAP (ours)	0.5358	0.2144	0.2802	0.0769

TABLE XIII

RGB RECONSTRUCTION BEFORE AND AFTER GAUSSIAN REFINEMENT ON FOUR LERF SCENES. BOTH INITIALIZERS USE THE SAME 30,000-ITERATION 3DGS REFINEMENT PROTOCOL. FF DENOTES THE FEED-FORWARD PREDICTION BEFORE REFINEMENT.

Scene	Tree (GeoBridgeNet-A3)		Flat encoding		Vanilla 3DGS PSNR
	FF PSNR	Refined PSNR	FF PSNR	Refined PSNR	
Figurines	14.79	24.29	14.98	21.24	24.10
Teatime	14.53	21.92	15.00	20.13	24.54
Ramen	16.56	24.69	16.58	22.95	24.75
Waldo	13.70	22.42	13.48	18.68	24.20
Mean	14.90	23.33	15.01	20.75	24.40

TABLE XIV

DOWNSTREAM OPEN-VOCABULARY 3-D SEMANTIC EVALUATION ON FOUR LERF SCENES. GAUSSIAN REPRESENTATIONS INITIALIZED BY THE TREE-DERIVED GEOBRIDGEBNET-A3 AND FLAT-ENCODING COUNTERPART ENTER THE SAME COS3D SEMANTIC PIPELINE.

Method	Mean		Figurines	Teatime	Ramen	Waldo
	mIoU $\uparrow$	mAcc $\uparrow$	mIoU $\uparrow$	mIoU $\uparrow$	mIoU $\uparrow$	mIoU $\uparrow$
COS3D (shallow MLPs) [14]	49.75	70.60	53.90	66.91	36.61	41.56
COS3D (kernel regression) [14]	50.76	72.08	60.03	65.07	35.86	42.10
Tree (GeoBridgeNet-A3) init. + COS3D	50.54	71.32	59.40	65.33	33.59	43.86
Flat encoding init. + COS3D	49.74	75.67	64.36	56.64	33.19	44.78

TABLE XV

PALETO-VARIANT ANALYSIS IN THE FOUR-SCENE COS3D TRANSFER PIPELINE. PARAMS COUNTS THE COMPLETE FEED-FORWARD GAUSSIAN INITIALIZER; THE SHARED SCENE-LEVEL 3DGS AND COS3D STAGES ARE EXCLUDED BECAUSE THE NUMBER OF OPTIMIZED GAUSSIANS IS SCENE DEPENDENT.

Variant	Params $\downarrow$ (M)	Refined PSNR (dB)	mIoU $\uparrow$ (%)	mAcc $\uparrow$ (%)	Acc@0.5 $\uparrow$ (%)
GeoBridgeNet-A1	52.4	23.46	51.06	70.58	53.67
GeoBridgeNet-A2	53.5	23.34	49.77	67.87	52.83
GeoBridgeNet-A3	62.0	23.33	50.54	71.32	53.00

REFERENCES

- [1] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [2] S. F. Bhat, I. Alhashim, and P. Wonka, "AdaBins: Depth estimation using adaptive bins," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4008–4017, doi: 10.1109/CVPR46437.2021.00400.
- [3] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "Neural window fully-connected CRFs for monocular depth estimation," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3906–3915, doi: 10.1109/CVPR52688.2022.00389.
- [4] D. Charatan, S. L. Li, A. Tagliasacchi, and V. Sitzmann, "pixel-Splat: 3D Gaussian splats from image pairs for scalable generalizable 3D reconstruction," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19457–19467, doi: 10.1109/CVPR52733.2024.01840.
- [5] Y. Chen, H. Xu, C. Zheng, B. Zhuang, M. Pollefeys, A. Geiger, T.-J. Cham, and J. Cai, "MVSplat: Efficient 3D Gaussian splatting from sparse multi-view images," in *Proc. European Conference on Computer Vision*, 2024, pp. 370–386, doi: 10.1007/978-3-031-72664-4_21.
- [6] H. Xu, S. Peng, F. Wang, H. Blum, D. Barath, A. Geiger, and M. Pollefeys, "DepthSplat: Connecting Gaussian splatting and depth," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 16453–16463, doi: 10.1109/CVPR52734.2025.01534.
- [7] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *Proc. International Conference on Learning Representations*, 2019.
- [8] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7029–7038, doi: 10.1109/CVPR.2019.00720.
- [9] H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu, "Hierarchical representations for efficient architecture search," in *Proc. International Conference on Learning Representations*, 2018.
- [10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002, doi: 10.1109/4235.996017.
- [11] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "TartanAir: A dataset to push the limits of visual SLAM," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 4909–4916, doi: 10.1109/IRoS45743.2020.9341801.
- [12] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "LERF: Language embedded radiance fields," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19672–19682, doi: 10.1109/ICCV51070.2023.01807.
- [13] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, Art. no. 139, 2023, doi: 10.1145/3592433.
- [14] R. Zhu, K.-H. Hui, Z. Liu, Q. Wu, W. Tang, S. Qiu, P.-A. Heng, and C.-W. Fu, "COS3D: Collaborative open-vocabulary 3D segmentation," in *Proc. Advances in Neural Information Processing Systems*, vol. 38, 2025.
- [15] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697–8710, doi: 10.1109/CVPR.2018.00907.
- [16] Z. Lu, I. Whalen, V. Boddeti, Y. Dhebar, K. Deb, E. Goodman, and W. Banzhaf, "NSGA-Net: Neural architecture search using multi-objective genetic algorithm," in *Proc. Genetic and Evolutionary Computation Conference*, 2019, pp. 419–427, doi: 10.1145/3321707.3321729.
- [17] J. Fang, Y. Sun, Q. Zhang, K. Peng, Y. Li, W. Liu, and X. Wang, "FNA++: Fast network adaptation via parametric remapping and architecture search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 2990–3004, 2021, doi: 10.1109/TPAMI.2020.3044416.
- [18] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei, "Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 82–92, doi: 10.1109/CVPR.2019.00017.
- [19] R. Liu, L. Wang, Z. Yu, H. Zhang, X. Liu, B. Sun, X. Huo, and J. Zhang, "SCTNet-NAS: Efficient semantic segmentation via neural architecture search for cloud-edge collaborative perception," *Complex & Intelligent Systems*, vol. 11, no. 8, Art. no. 365, 2025, doi: 10.1007/s40747-025-01996-5.
- [20] L. Huynh, P. Nguyen, J. Matas, E. Rahtu, and J. Heikkilä, "Lightweight monocular depth with a novel neural architecture search method," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 326–336, doi: 10.1109/WACV51458.2022.00040.
- [21] Z. Yu, H. Zhang, R. Liu, S. Dai, X. Chen, W. Sheng, and Y. Jin, "Mde-EvoNAS: Automatic network architecture design for monocular depth estimation via evolutionary neural architecture search," *Swarm and Evolutionary Computation*, vol. 93, Art. no. 101837, 2025, doi: 10.1016/j.swevo.2024.101837.
- [22] H. Zhang, Z. Yu, Y. Jin, X. Liu, W. Sheng, R. Liu, X. Li, Q. Liu, and R. Cheng, "Efficient evolutionary neural architecture search with hierarchical parameter mapping for monocular depth estimation," *IEEE Transactions on Evolutionary Computation*, vol. 30, no. 5, pp. 1893–1907, 2026, doi: 10.1109/TEVC.2025.3614261.
- [23] H. Zhang, Q. Yao, J. T. Kwok, and X. Bai, "Searching a high performance feature extractor for text recognition network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 6231–6246, 2023, doi: 10.1109/TPAMI.2022.3205748.
- [24] Y. Tang, M. Liu, B. Li, Y. Wang, and W. Ouyang, "NAS-PED: Neural architecture search for pedestrian detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 3, pp. 1800–1817, 2025, doi: 10.1109/TPAMI.2024.3507918.
- [25] X. Guo, Z. Zhu, T. Yang, B. Lin, J. Huang, J. Deng, G. Huang, J. Zhou, and J. Lu, "Gait recognition in the wild: A large-scale benchmark and NAS-based baseline," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 6, pp. 4535–4552, 2025, doi: 10.1109/TPAMI.2025.3546482.
- [26] Z. Zhong, X. Liu, J. Jiang, D. Zhao, and S. Wang, "Dual-level cross-modality neural architecture search for guided image super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 9, pp. 8249–8267, 2025, doi: 10.1109/TPAMI.2025.3578468.
- [27] Z. Lu, G. Sreekumar, E. Goodman, W. Banzhaf, K. Deb, and V. N. Boddeti, "Neural architecture transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 2971–2989, 2021, doi: 10.1109/TPAMI.2021.3052758.
- [28] Y. Duan, X. Chen, H. Xu, Z. Chen, X. Liang, T. Zhang, and Z. Li, "TransNAS-Bench-101: Improving transferability and generalizability of cross-task neural architecture search," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5247–5256, doi: 10.1109/CVPR46437.2021.00521.
- [29] X. Zhou, X. Wu, L. Feng, Z. Lu, and K. C. Tan, "Design principle transfer in neural architecture search via large language models," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 39, no. 21, pp. 23000–23008, 2025, doi: 10.1609/aaai.v39i21.34463.

- [30] C. White, W. Neiswanger, S. Nolen, and Y. Savani, "A study on encodings for neural architecture search," in *Proc. Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [31] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 394–407, 2020, doi: 10.1109/TEVC.2019.2916183.
- [32] X. Chen, L. Xie, J. Wu, and Q. Tian, "Progressive DARTS: Bridging the optimization gap for NAS in the wild," *International Journal of Computer Vision*, vol. 129, no. 3, pp. 638–655, 2021, doi: 10.1007/s11263-020-01396-x.
- [33] H. Kitano, "Designing neural networks using genetic algorithms with graph generation system," *Complex Systems*, vol. 4, no. 4, pp. 461–476, 1990.
- [34] F. Gruau, "Neural network synthesis using cellular encoding and the genetic algorithm," Ph.D. dissertation, Université Claude Bernard Lyon 1, Lyon, France, 1994.
- [35] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002, doi: 10.1162/106365602320169811.
- [36] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci, "A hypercube-based encoding for evolving large-scale neural networks," *Artificial Life*, vol. 15, no. 2, pp. 185–212, 2009, doi: 10.1162/artl.2009.15.2.15202.
- [37] C. Broni-Bediako, Y. Murata, L. H. B. Mormille, and M. Atsumi, "Evolutionary NAS with gene expression programming of cellular encoding," in *Proc. IEEE Symposium Series on Computational Intelligence*, 2020, pp. 2670–2676, doi: 10.1109/SSCI47803.2020.9308346.
- [38] J. A. Fuentes-Tomás, E. Mezura-Montes, H. G. Acosta-Mesa, and A. Márquez-Grajales, "Tree-based codification in neural architecture search for medical image segmentation," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 3, pp. 597–607, 2024, doi: 10.1109/TEVC.2024.3353182.
- [39] S. Yang, Y. Tian, X. Xiang, S. Peng, and X. Zhang, "Accelerating evolutionary neural architecture search via multifidelity evaluation," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 4, pp. 1778–1792, 2022, doi: 10.1109/TCDS.2022.3179482.
- [40] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun, "Single path one-shot neural architecture search with uniform sampling," in *Proc. European Conference on Computer Vision*, 2020, pp. 544–560, doi: 10.1007/978-3-030-58517-4_32.
- [41] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han, "Once-for-all: Train one network and specialize it for efficient deployment," in *Proc. International Conference on Learning Representations*, 2020.
- [42] C. Li, S. Lin, T. Tang, G. Wang, M. Li, X. Liang, and X. Chang, "Boss-NAS family: Block-wisely self-supervised neural architecture search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 5, pp. 3500–3514, 2025, doi: 10.1109/TPAMI.2025.3529517.
- [43] L. Yan, Y. Du, J. Liang, B. Qu, C. Li, and K. Yu, "A weight inheritance and guidance strategy-based evolutionary network architecture search," *IEEE Transactions on Evolutionary Computation*, vol. 30, no. 2, pp. 549–563, 2026, doi: 10.1109/TEVC.2025.3558334.
- [44] Y. Sun, H. Wang, B. Xue, Y. Jin, G. G. Yen, and M. Zhang, "Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 350–364, 2020, doi: 10.1109/TEVC.2019.2924461.
- [45] L. Ma, N. Li, G. Yu, X. Geng, S. Cheng, X. Wang, M. Huang, and Y. Jin, "Pareto-wise ranking classifier for multiobjective evolutionary neural architecture search," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 3, pp. 570–581, 2024, doi: 10.1109/TEVC.2023.3314766.
- [46] B. Cao, X. Luo, X. Liu, and Y. Li, "Comprehensive-forecast multi-objective genetic programming for neural architecture search," *IEEE Transactions on Evolutionary Computation*, vol. 30, no. 1, pp. 76–90, 2026, doi: 10.1109/TEVC.2025.3570195.
- [47] N. Li, B. Xue, L. Ma, and M. Zhang, "Automatic fuzzy architecture design for defect detection via classifier-assisted multiobjective optimization approach," *IEEE Transactions on Evolutionary Computation*, vol. 30, no. 2, pp. 479–490, 2026, doi: 10.1109/TEVC.2025.3530416.
- [48] N. Lee, T. Ajanthan, and P. H. S. Torr, "SNIP: Single-shot network pruning based on connection sensitivity," in *Proc. International Conference on Learning Representations*, 2019.
- [49] C. Wang, G. Zhang, and R. Grosse, "Picking winning tickets before training by preserving gradient flow," in *Proc. International Conference on Learning Representations*, 2020.
- [50] J. Mellor, J. Turner, A. Storkey, and E. J. Crowley, "Neural architecture search without training," in *Proc. International Conference on Machine Learning*, vol. 139, 2021, pp. 7588–7598.
- [51] M. S. Abdelfattah, A. Mehrotra, Ł. Dudziak, and N. D. Lane, "Zero-cost proxies for lightweight NAS," in *Proc. International Conference on Learning Representations*, 2021.
- [52] J. Lee and B. Ham, "AZ-NAS: Assembling zero-cost proxies for network architecture search," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5893–5903, doi: 10.1109/CVPR52733.2024.00563.
- [53] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al., "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024.
- [54] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, J. Jiao, and Y. Liu, "VMamba: Visual state space model," in *Proc. Advances in Neural Information Processing Systems*, vol. 37, 2024.
- [55] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth Anything V2," in *Proc. Advances in Neural Information Processing Systems*, vol. 37, 2024.
- [56] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. European Conference on Computer Vision*, 2012, pp. 746–760, doi: 10.1007/978-3-642-33715-4_54.
- [57] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361, doi: 10.1109/CVPR.2012.6248074.
- [58] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," *ACM Transactions on Graphics*, vol. 37, no. 4, Art. no. 65, 2018, doi: 10.1145/3197517.3201323.
- [59] Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, and J. Sun, "DetNAS: Backbone search for object detection," in *Proc. Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [60] X. Dong and Y. Yang, "Searching for a robust neural architecture in four GPU hours," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1761–1770, doi: 10.1109/CVPR.2019.00186.
- [61] J. H. Lee, M. K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *arXiv preprint arXiv:1907.10326*, 2019.
- [62] Z. Li, Z. Chen, X. Liu, and J. Jiang, "DepthFormer: Exploiting long-range correlation and local information for accurate monocular depth estimation," *Machine Intelligence Research*, vol. 20, no. 6, pp. 837–854, 2023, doi: 10.1007/s11633-023-1458-0.
- [63] S. F. Bhat, I. Alhashim, and P. Wonka, "LocalBins: Improving depth estimation by learning local distributions," in *Proc. European Conference on Computer Vision*, 2022, pp. 480–496, doi: 10.1007/978-3-031-19769-7_28.
- [64] V. Patil, C. Sakaridis, A. Liniger, and L. Van Gool, "P3Depth: Monocular depth estimation with a piecewise planarity prior," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1600–1611, doi: 10.1109/CVPR52688.2022.00166.
- [65] L. Piccinelli, C. Sakaridis, and F. Yu, "iDisc: Internal discretization for monocular depth estimation," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21477–21487, doi: 10.1109/CVPR52729.2023.02057.
- [66] S. Shao, Z. Pei, X. Wu, Z. Liu, W. Chen, and Z. Li, "IEBins: Iterative elastic bins for monocular depth estimation," in *Proc. Advances in Neural Information Processing Systems*, vol. 36, 2023.
- [67] Z. Li, X. Wang, X. Liu, and J. Jiang, "BinsFormer: Revisiting adaptive bins for monocular depth estimation," *IEEE Transactions on Image Processing*, vol. 33, pp. 3964–3976, 2024, doi: 10.1109/TIP.2024.3416065.
- [68] A. Agarwal and C. Arora, "Attention attention everywhere: Monocular depth prediction with skip attention," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 5850–5859, doi: 10.1109/WACV56688.2023.00581.
- [69] S. Kim, J. Kang, D. Kim, and S. Lee, "CLIP can understand depth," *Pattern Recognition*, vol. 172, Art. no. 112475, 2026, doi: 10.1016/j.patcog.2025.112475.
- [70] S. Li, H. Bai, Y. Gao, C. Lv, H. Yuan, C. Li, W. Hua, and T. Xie, "LiftFormer: Lifting and frame theory based monocular depth estimation using depth and edge oriented subspace representation," *IEEE Transactions on Multimedia*, vol. 27, pp. 9098–9110, 2025, doi: 10.1109/TMM.2025.3613146.
- [71] A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter, and V. Koltun, "Depth Pro: Sharp monocular metric depth in less than a second," in *Proc. International Conference on Learning Representations*, 2025.

- [72] K. Wang, Z. Yan, J. Fan, W. Zhu, X. Li, J. Li, and J. Yang, "DCDepth: Progressive monocular depth estimation in discrete cosine domain," in *Proc. Advances in Neural Information Processing Systems*, vol. 37, 2024.
- [73] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen, "Metric3D v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10579–10596, 2024, doi: 10.1109/TPAMI.2024.3444912.
- [74] H. Yu, H. Lin, J. Wang, J. Li, Y. Wang, X. Zhang, Y. Wang, X. Zhou, R. Hu, and S. Peng, "InfiniDepth: Arbitrary-resolution and fine-grained depth estimation with neural implicit fields," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 26920–26930.
- [75] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12159–12168, doi: 10.1109/ICCV48922.2021.01196.
- [76] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelNeRF: Neural radiance fields from one or few images," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4576–4585, doi: 10.1109/CVPR46437.2021.00455.
- [77] M. Suhail, C. Esteves, L. Sigal, and A. Makadia, "Generalizable patch-based neural rendering," in *Proc. European Conference on Computer Vision*, 2022, pp. 156–174, doi: 10.1007/978-3-031-19824-3_10.
- [78] Y. Du, C. Smith, A. Tewari, and V. Sitzmann, "Learning to render novel views from wide-baseline stereo pairs," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4970–4980, doi: 10.1109/CVPR52729.2023.00481.
- [79] H. Xu, A. Chen, Y. Chen, C. Sakaridis, Y. Zhang, M. Pollefeys, A. Geiger, and F. Yu, "MuRF: Multi-baseline radiance fields," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20041–20050, doi: 10.1109/CVPR52733.2024.01894.
- [80] W. Long, H. Wu, S. Jiang, J. Zhang, X. Ji, and S. Gu, "IDESplat: Iterative depth probability estimation for generalizable 3D Gaussian splatting," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 33248–33258.
- [81] H. Tanaka, D. Kunin, D. L. K. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," in *Proc. Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [82] M. Lin, P. Wang, Z. Sun, H. Chen, X. Sun, Q. Qian, H. Li, and R. Jin, "Zen-NAS: A zero-shot NAS for high-performance image recognition," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2021, pp. 337–346, doi: 10.1109/ICCV48922.2021.00040.
- [83] Z. Zhang and Z. Jia, "GradSign: Model performance inference with theoretical insights," in *Proc. International Conference on Learning Representations*, 2022.

Rui Wang is an undergraduate student with the School of Information Science and Technology, Hangzhou Normal University, Hangzhou, China. His research interests include computer vision and neural architecture search, with a focus on efficient visual model design, neural network architecture optimization, model optimization for visual tasks, and deep learning methods for practical applications.

Haoyu Zhang received the Ph.D. degree in automatic control from Donghua University, Shanghai, China, in 2022. He is currently an Associate Professor at the Hangzhou Normal University, and a Postdoctoral Researcher at Henan University. He has also been a visiting scholar at Westlake University and Hong Kong Polytechnic University. With over twenty publications as first or co-author, his research focuses on computer vision, evolutionary learning, neural architecture search, and depth estimation & 3D reconstruction, with works appearing in top-tier journals and conferences like IEEE Transactions on Evolutionary Computation, ACM Multimedia, and NeurIPS.

Ran Cheng (Senior Member, IEEE) received the B.Sc. degree from the Northeastern University, Shenyang, China, in 2010, and the Ph.D. degree from the University of Surrey, Guildford,

U.K., in 2016. He is currently an Associate Professor with the Department of Data Science and Artificial Intelligence, The Hong Kong Polytechnic University, Hong Kong SAR, China.

He is a recipient of the IEEE Transactions on Evolutionary Computation Outstanding Paper Award (2018 and 2021), the IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award (2019), the IEEE Computational Intelligence Magazine Outstanding Paper Award (2020), and the IEEE Computational Intelligence Society Early Career Award (2025). He is the Founding Chair of the IEEE Computational Intelligence Society Shenzhen Chapter. He is an Associate Editor of IEEE Transactions on Evolutionary Computation, IEEE Transactions on Artificial Intelligence, IEEE Transactions on Emerging Topics in Computational Intelligence, and IEEE Transactions on Cognitive and Developmental Systems.