GENERALIST MODELS · SPATIAL INTELLIGENCE

Iris-Z: Scaling One Generative Model Past Specialist Geometry Models

Alejandro Cañada[†], **Chema Garabito**

SperidLabs · [†]Project lead · Code and models: github.com/speridlabs/iris-z-sdk

ABSTRACT

Language moved from many task-specific models to one unified model: a single generative model with a single generative training task that handles all of them within the same backbone, and keeps improving with data, parameters and compute. We believe the bitter lesson applies just as much to computer vision, 3D and spatial understanding, in the end to visual intelligence: the effort should go into general architectures that solve all these tasks in a unified way, and into scaling them, rather than into task-specific designs. We test this on monocular geometry, a task that has belonged to specialised architectures for years. Iris-Z is a general image generator that produces appearance and geometry together. A single backbone denoises, reasons about and understands the image and its depth at the same time, and one set of weights generates images with their depth, estimates depth from photographs and paints images onto a given depth. At enough scale it surpasses state-of-the-art specialists on established depth benchmarks while keeping the generative abilities of the backbone it started from without specialized layers. We release two models for the community.

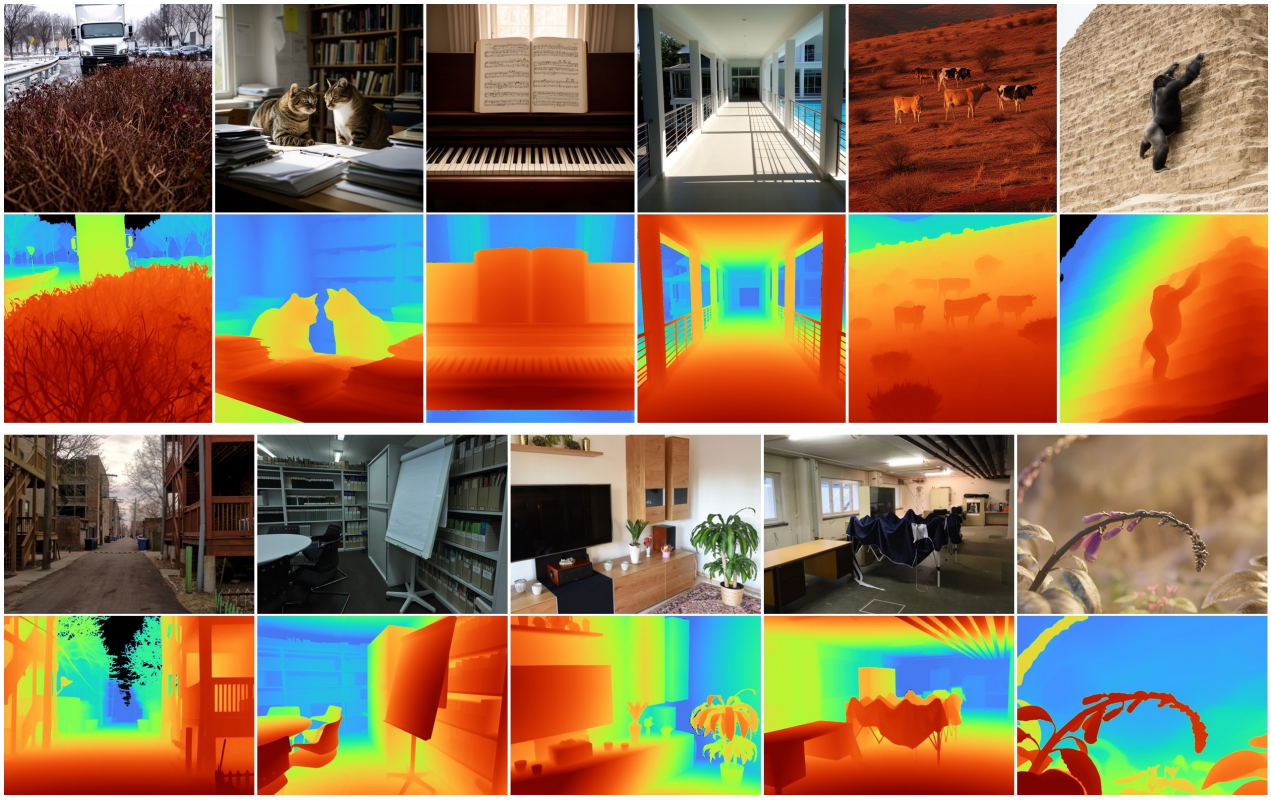


FIG. 1 Top: prompt → image and depth, generated together. Bottom: photographs → depth. All from Iris-Z-big. Depth is coloured on log

distance, near = red, far = blue; black: no depth or sky.

01 · Introduction

Language went from many task-specific models to one unified model that handles all of them within the same backbone: a single generative model with a single generative training task, simple and foundational enough to cover all the others. Trained to predict the next token on enough data, it learned to translate, summarise and answer questions at a level competitive with models built for each of those tasks [1]. And the same single model kept getting better at every one of those tasks as data, parameters and compute grew [2, 3], giving one way to scale every text task forward at once. We believe this is the bitter lesson in its plainest form: general methods that scale with computation end up ahead of task-specific methods [4].

We believe the bitter lesson applies to computer vision, 3D and spatial understanding, and in the end to visual intelligence, as much as it applied to language. The effort should go into finding general architectures that solve all of these fields and tasks in a unified way, and then into scaling that one model as far as possible, rather than into task-specific designs. Visual intelligence is now following this path through visual generative models. For example, Vision Banana instruction-tunes the Nano Banana Pro image generator on a small amount of task data and matches or beats specialists such as Segment Anything 3 and Depth Anything without losing its ability to generate [6]. Veo 3 segments objects, finds edges and solves simple visual tasks without being trained for any of them [5]. Models trained to both understand and generate images with the same weights such as BAGEL [7] and PixelUMM [8] also point in a similar direction.

Data is the other half of this argument. General generative models can be trained on massive amounts of visual data, web-scale images and videos that need no labels designed for one task. A unified architecture such as Iris-Z, where each modality is one more group of tokens in the same sequence, lets more modalities join that training. In our experiments, the larger the pre-trained generator and the more data it is trained on, the better it does on task-specific benchmarks (Fig. 2b): parameter count, compute and above all scalable pre-training data are what move the result. The more compute and data go into a single architecture, the better it becomes at each specific task, which gives one way to scale every visual task at once.

Monocular geometry is a hard place to test this. The reference model, MoGe-3 [11], is a task-specific ViT that as of today leads the benchmarks. In MoGe-3's own comparison, models like Apple's Depth Pro [12], ByteDance's Depth Anything 3 [13], UniDepth V2 or UniK3D all stay behind it. Image generators that are repurposed as depth estimators like the Marigold family [14, 16], although strong, still perform worse than MoGe-3 and also than MoGe-2 when evaluated on MoGe-3's held-out test sets. Vision Banana reports a lower metric-depth error than MoGe-2 [10], but it predates MoGe-3 and is not available to test.

We built Iris-Z to test this idea and to show that a general model can beat models designed for one task. We took a general image generator and taught it not only to generate appearance (RGB) but to generate and reason about geometry (depth) jointly with it, denoising both together as one RGBD output. In Iris-Z appearance and geometry are produced together, and a single backbone denoises, reasons about and understands both at once. Our view is that a generative model should generate and understand the world in all its modalities jointly, and that the geometry it produces along with an image is part of what it knows about the scene and the world.

We see this as a path forward to unifying visual understanding. The same recipe extends to further modalities, such as surface normals, segmentation, motion or the camera, each added as one more group of tokens in the same sequence, so that a single model learns all of them together and improves on all of them as it scales.

In our experiments generalist image models have proven better than specialists with enough scale. Iris-Z-big is, to our knowledge, the first open general-purpose generator to beat MoGe-3 on Marigold's benchmarks, and it is also ahead of MoGe-3 on average on MoGe-3's own benchmarks at the resolutions it was trained for, while still being a general generative model that does the same tasks it did before we re-trained it. Each step of scale we added, more data, a full fine-tune and a larger backbone, improved the result (Fig. 2b) without the need for a depth-specific layer. We take this as a sign that generative AI, computer vision and spatial intelligence are converging on one model that has all of these capabilities and scales better than today's specialised systems.

In summary, we make the following contributions:

- Iris-Z, a generative model that denoises image and depth jointly in one sequence and serves four tasks with one set of weights, released in two sizes, Iris-Z-big and Iris-Z-small.

– State-of-the-art affine-invariant depth performance on Marigold's and MoGe-3's benchmarks (Sec. 6).

– Evidence that the improvement in depth estimation comes from scale rather than task design. Across three runs we changed only the training data and the backbone size, and each run lowers the depth error (Sec. 6).

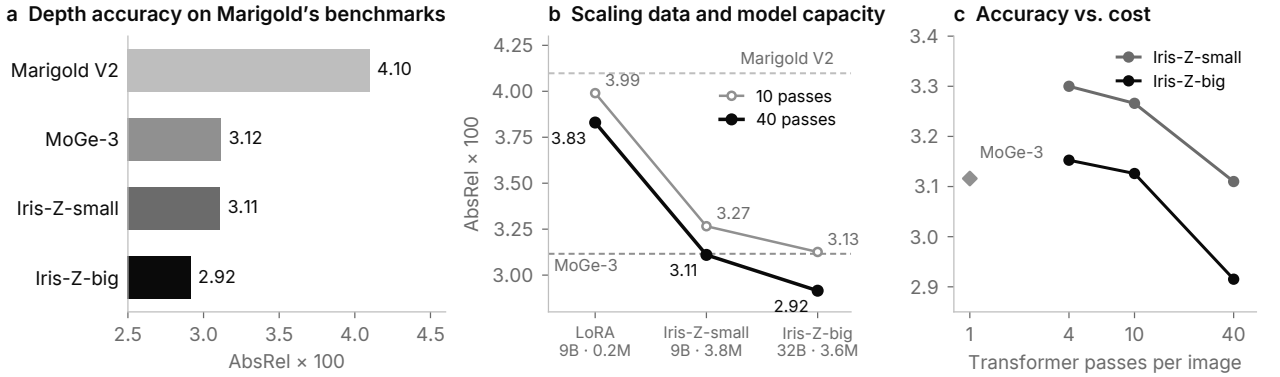


FIG. 2 Iris-Z on Marigold V2's benchmarks, mean AbsRel×100 (lower is better). **a**: Our two models vs Marigold V2 and MoGe-3. **b**: scaling: a LoRA on Iris-Z-small's backbone that saw 0.2M frames, a full fine-tune of it (Iris-Z-small) that saw 3.8M unique frames, then Iris-Z-big, a backbone 3.5× larger, that saw 3.6M. Black: 40 inference passes, the sampler we report; grey: 10 passes. **c**: accuracy against the number of DiT passes per image (4: one seed, 10: 2 steps from 5 seeds, 40: 4 steps from 10 seeds). Iris-Z-big's 10-pass points are from an earlier checkpoint of the same training.

02 · Related Work

Monocular depth specialists. Learning depth from a single image began with networks regressed on one dataset at a time, with a scale-invariant loss to deal with the ambiguity of a single view [23]. MiDaS showed that mixing many datasets with different scales trained with a loss invariant to scale and shift transferred to unseen domains far better than any single source [24]. DPT moved the same recipe to a vision transformer [25]. Metric depth needed camera-aware approaches like ZoeDepth which adds metric heads to a relative model [26], Metric3D that maps every image to a canonical camera [27] or UniDepth that predicts the camera together with the depth [28]. Depth Anything scaled the training data instead, pseudo-labelling about 62M unlabelled images with a teacher [29]; its second version trains that teacher on synthetic data only [30]. Depth Pro then focused on sharp boundaries and focal length estimation [12], and Depth Anything 3 extended the approach to multiple views [13]. MoGe predicts an affine-invariant point map with a robust alignment in training [9], MoGe-2 adds metric scale [10], and MoGe-3 refines the point map in 3D with sparse convolutions [11]. This line of work improved by finding better depth-specific strategies and more varied and better labelled data, while each model stayed task-specific.

Generative priors for dense prediction. Marigold showed that a text-to-image latent diffusion model fine-tuned on a small amount of synthetic data becomes a strong zero-shot depth estimator [14]. Much of the follow-up work made this faster: DepthFM replaces diffusion with flow matching from image to depth [32], Lotus predicts the clean target in a single step [33], and Martin Garcia et al. show that a single-step model matches the multi-step one [15]. Other work extended it to new outputs and backbones. GeoWizard predicts depth and normals together [31], DepthCrafter applies the approach to a video generator [34], Marigold V2 moves to diffusion transformers [16], and Iris-3B compares pixel and latent-space generators [17]. In all of these the generator becomes an image-to-depth model. The image is always the input, and the ability to generate images is lost along the way. Iris-Z keeps the generative objective and text-to-image generation, and depth estimation is one of several tasks the same joint model performs.

Joint RGB-D and multi-modal generation. A smaller line of work produces geometry together with appearance. LDM3D trains one latent diffusion model to output an image and its depth from a prompt [35], while JointNet keeps the text-to-image model frozen and adds a parallel branch for a dense modality such as depth [36]. Wonder3D extends this across views, producing colours and normals with cross-domain attention [37], and Aether trains a video model to predict geometry alongside the frames [38]. None of these were shown to reach specialist accuracy on established depth benchmarks, which is the gap Iris-Z addresses.

Generalist vision through generation. Visual in-context learning shows a model an example input and its output side by side together with a new input, and asks it to fill in the missing output, so the task is defined by the example rather than by the architecture. Bar et al. do this with an inpainting model on a grid of images [39], and Painter trains a masked image model on many tasks in this format [40]. InstructCV instruction-tunes a text-to-image model to perform vision tasks named in text [41], and OmniGen trains one generator for editing, conditional generation and several classic vision tasks [42]. More recently, Vision Banana showed that a frontier image generator with instruction-tuning matches specialists on segmentation and depth [6], and Veo 3 solves visual tasks it was never trained for [5]. Unified multimodal models go further and both read and write images in one network. Transfusion combines next-token prediction and diffusion in one transformer [43], Janus uses separate visual encoders for understanding and generation [44], and BAGEL and PixelUMM train such models at larger scale [7, 8].

Scaling visual generators. Diffusion transformers improve predictably with compute [45], and rectified-flow transformers show smooth scaling of validation loss that tracks human preference up to billions of parameters [46]. Whether the geometry a generator learns scales the same way is less studied. Our scaling path (Fig. 2b) is a first data point for the depth modality.

03 · Method

Iris-Z comes from a pretrained text-to-image diffusion transformer and the only new parameters we add belong to a learned modality embedding. There is no depth head and no second network. Depth is encoded as an image by the same autoencoder that encodes the image and placed in the same token sequence, and training uses the base model's flow-matching loss together with a loss on the decoded depth. Below we describe how image and depth share one sequence (Fig. 3), how depth is represented, which tasks are trained, and how we sample. The backbones, training settings and data are in Sec. 4.

Image and depth in one sequence. The image and its depth are both encoded as images by the backbone's autoencoder, and their latents form one token sequence together with the prompt. Each token carries three embeddings that identify it. The *spatial position embedding* (the backbone's rotary row and column) is the same for the image token and the depth token of a patch, so the model knows they describe the same point of the scene. The *sequence position embedding*, an extra rotary coordinate the backbone already uses to tell several input images apart, is 0 for image tokens and 1 for depth tokens. A learned *modality embedding* labels each token as generated image, generated depth, given image or given depth. Image and depth each carry their own *flow-matching time* t: whichever is generated follows the rectified-flow objective of the base model [20, 46] at its own t, and whichever is given is fed clean and modulated with $t = 0$. Because attention runs over the whole sequence, image and depth condition each other at every step of denoising.

Depth as an image. Depth is stored as per-image normalised log ray distance. The 2nd to 98th percentile of an image's log distance is mapped to $[-1, 0.95]$, so every image uses the same range and the prediction is defined up to a scale and shift in log space. Sky is set to $+1$, which leaves a margin between the farthest real depth and the sky. The value is copied into three channels and encoded by the same autoencoder as the image, without changes; decoding averages the three channels. At inference, pixels above 0.975 are marked as sky.

Do multiple tasks with the same weights. Each training sample draws one task. *Joint*: image and depth are noised to the same time and both are scored (prompt $\rightarrow$ RGBD). *Image given*: only depth is denoised (image $\rightarrow$ depth). *Depth given*: the reverse (depth $\rightarrow$ image). *Independent*: image and depth are noised to independent times. *Image only*: the depth stays pure noise and only the image is scored (prompt $\rightarrow$ image); this keeps training plain text-to-image generation and prevents the model from drifting away from the base model's images. With this mix of tasks, one set of weights serves four tasks at inference. A second loss decodes the depth the model currently predicts and compares it with the target in pixel space. Captions are dropped for a share of samples, so image $\rightarrow$ depth works without a prompt.

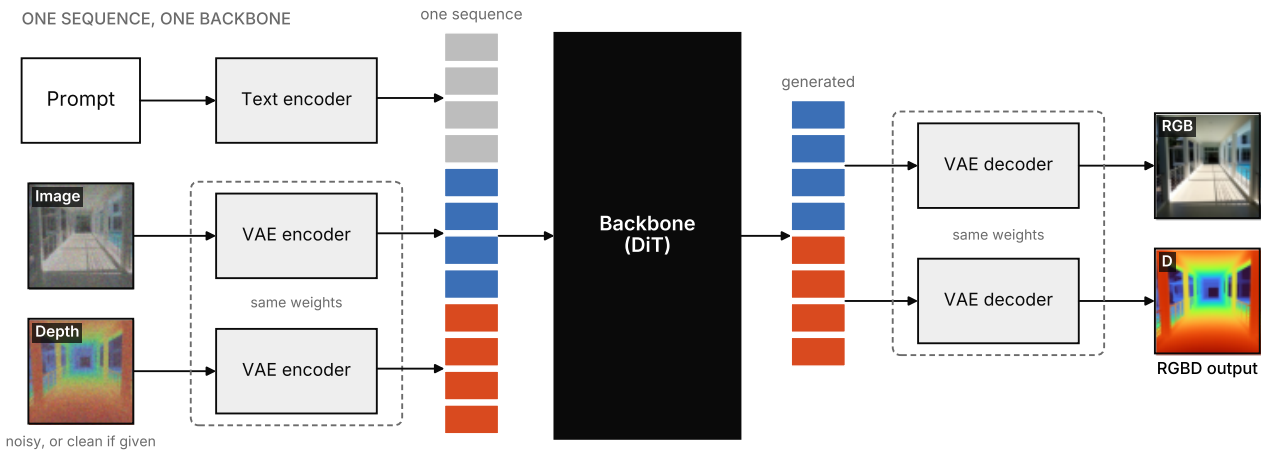


FIG. 3 Iris-Z. The prompt, the image and the depth are encoded into one token sequence and processed by a single DiT. Image and depth share the VAE encoder and decoder. Image and depth enter as noisy latents, or clean when the task gives them.

Sampling depth. Image → depth denoises the depth with the image given and an empty prompt, so both branches of classifier-free guidance are identical and each step costs one transformer pass. Sampling starts from noise, so we combine a few short trajectories from different seeds (Sec. 4.4).

04 · Implementation Details

4.1 Backbones

We use two open text-to-image transformers from the FLUX.2 family as the backbone [18, 19]: Iris-Z-big starts from FLUX.2 [dev] (32B parameters, Mistral Small 24B text encoder) and Iris-Z-small from FLUX.2 [klein] Base 9B (Qwen3-8B text encoder). Both are full fine-tunes of the transformer; the autoencoder and the text encoder stay frozen. FLUX.2 [dev] is guidance-distilled: we remove its guidance embedding and train classifier-free guidance back through caption dropout. The only new parameters are the modality embedding, initialised at zero, so training starts from the unchanged base model.

4.2 Training

Both models are trained with AdamW in bf16 mixed precision and with sharded data parallelism. The decoded-depth loss takes the clean depth latent implied by the prediction, $\hat{x}_0 = x_t - t\,v$, decodes it with the frozen autoencoder and computes an L1 to the target over valid pixels, weighted by $(1 - t)$ so that near-clean steps count most; it is added with weight 1 to the flow-matching loss of every sample whose depth is scored. On sources with sparse or holed depth (LiDAR, stereo, relative reconstructions) only this decoded loss supervises depth, on measured pixels. Iris-Z-big keeps headroom for sky: real depth is squeezed into $[-1, 0.95]$ and sky alone is $+1$; Iris-Z-small uses the full range. We keep two exponential moving averages of the weights; the slower one is better on every benchmark for both models and is the one we release.

Iris-Z-big trains at a constant learning rate and is then annealed linearly to zero over its last 750 steps; Iris-Z-small keeps a constant learning rate. Both start at about 512^2 pixels and move a growing share of samples to about 1024^2, ending at 42% (Iris-Z-big) and 56% (Iris-Z-small).

4.3 Data

Each model was trained on about 3 to 4 million unique RGB-D frames, drawn from a pool of real indoor captures, driving scenes with LiDAR and stereo, multi-view reconstructions, rendered scenes and generated images. Every source was converted to metric ray distance. Sensor holes and sparse LiDAR stay as holes: on those frames depth is supervised only on measured pixels. Sky follows one rule per source and is stored as a fixed far value. Every frame has a dense caption, used as the prompt. Iris-Z-small saw about 3.8M unique frames, Iris-Z-big about 3.6M. Images are stored at about 512^2 and 1024^2 pixels. Validation frames are held out per source, split by scene. We do not train on the benchmark datasets.

4.4 Sampling and generation

Image → depth runs 4 steps from each of 10 seeds. Each seed's prediction is the average of the clean-depth estimates $\hat{x}_0 = x_t - t\,v$ from its last two steps. The seeds are aligned to a reference by least-squares scale and shift in log space and reduced by the per-pixel median: 40 transformer passes per image. Images under 0.5 MP run at about 512^2 pixels, larger

ones at about 1024^2 (with one exception on MoGe-3's benchmarks, Sec. 5). The benchmark then fits its own alignment to the result, as for any model.

Prompt $\rightarrow$ image and prompt $\rightarrow$ RGBD use 50 steps with classifier-free guidance (3.5 for Iris-Z-big, 4.0 for Iris-Z-small). For prompt $\rightarrow$ RGBD the depth of the joint generation can be used as is, or re-estimated from the finished image with the image $\rightarrow$ depth sampler, which is sharper.

Depth $\rightarrow$ image denoises the image with the depth given clean, starting from Gaussian noise: 50 steps from one seed with classifier-free guidance between the caption and the empty prompt. Without a prompt there is no guidance and each step is one transformer pass. Holes in the given depth are filled with the nearest measured value.

4.5 Generation is preserved

A depth estimator that forgets how to draw would be just another specialist. We generate with fixed held-out prompts and seeds and compare with the untouched base model on the same prompts and seeds (Tab. 1, Fig. 4) to ensure that Iris-Z is in fact a general model.

Prompt adherence holds as CLIP stays within 0.4 of the base model, slightly above it for Iris-Z-small. Images keep the base model's subject and composition (similarity 0.88–0.89). Preference scores drop (HPSv2.1 24.3 against 26.7 for Iris-Z-big) as the models drift a little towards the look of their training data. That same drift makes Iris-Z-big's images more photographic (0.48 against 0.38).

TAB. 1 Prompt $\rightarrow$ image against the untouched base model, same prompts and seeds, 50 steps with guidance. CLIP [21]: 100$\times$ the image–prompt cosine. Sim.: CLIP cosine to the base model's image (1 = identical). Photoreal: CLIP's zero-shot probability that the image is a photograph rather than a render or drawing. HPSv2.1 [22]: a learned human-preference score.

MODEL	CLIP	SIM.	PHOTOREAL	HPSV2.1
FLUX.2 [klein]	27.46	–	0.41	24.50
Iris-Z-small	27.74	0.875	0.34	24.01
FLUX.2 [dev]	27.85	–	0.38	26.71
Iris-Z-big	27.42	0.888	0.48	24.30



FIG. 4 Same prompt and seed. Top: the untouched base model. Bottom: Iris-Z, the image from a joint RGBD generation.

05 · Evaluation

Benchmarks. Two tables, each run exactly as its paper defines it. *Marigold's*: NYUv2, KITTI, ETH3D, ScanNet and DIODE with Marigold V2's split lists and masks (3,331 images), scored after RANSAC scale-and-shift alignment in log depth. *MoGe-3's*: NYUv2, KITTI, ETH3D, iBims-1, Sintel, DIODE and HAMMER with MoGe's masks (4,470 images), scored after MoGe's relative-error L1 alignment. We report AbsRel$\times$100.

Sets and resolution. We leave out two of MoGe-3's nine sets, Spring (2 MP) and Synth4K (8 MP): both models were trained at up to about 1024^2 pixels, and these sets test resolutions we did not train for. ETH3D, also high resolution, stays because it is one of Marigold V2's benchmarks too. On Marigold's table we keep the rule of Sec. 4.4 (under 0.5 MP at about 512^2, larger at about 1024^2). On MoGe-3's table, sets with sparse LiDAR ground truth (KITTI) also run at about 1024^2.

Baselines. Marigold V2 (a single-step model by design, one pass) and MoGe-3 (ViT-G, resolution level 9, 3 refinement steps) are run by our code on the same pixels.

06 · Results

MoGe-3's protocol. On MoGe-3's own benchmarks and alignment (Tab. 3, Fig. 5) Iris-Z-big is the most accurate model on average (3.56 against 3.70 for MoGe-3) and on six of the seven sets; MoGe-3 keeps DIODE. Iris-Z-small is ahead of MoGe-3 on NYUv2 and HAMMER and behind on average (3.84).

Marigold's protocol. Iris-Z-big is the most accurate model on Marigold's benchmarks (Tab. 4): 2.92, against 3.12 for MoGe-3 and 4.10 for Marigold V2. It is best on NYUv2, ETH3D and ScanNet; MoGe-3 keeps KITTI and DIODE. Iris-Z-small, at 9B, is level with MoGe-3 (3.11).

Note that the estimates on the datasets that appear in both protocols (NYUv2, KITTI, ETH3D, DIODE) differ because each protocol prepares the data differently: Marigold's uses the native images and cameras with fixed depth ranges and crops, while MoGe's re-renders every sample into a centred, undistorted pinhole view (cropping or zooming some sets).

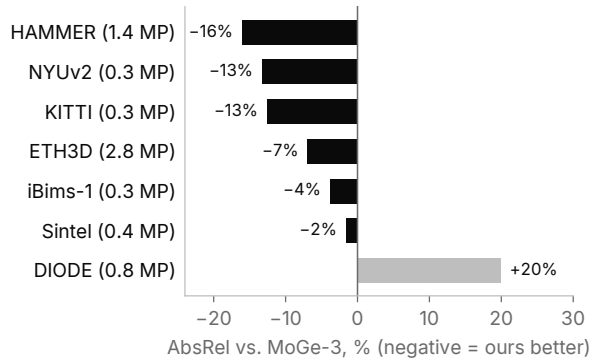


FIG. 5 Iris-Z-big against MoGe-3 on MoGe-3's benchmarks and alignment, per set. Black: ours better.

Scale. The same recipe was run three times with more scale each time (Fig. 2b). A LoRA that saw 0.2M frames scored 3.83 with 40 passes, already ahead of Marigold V2. A full fine-tune that saw about 3.8M unique frames, Iris-Z-small, brought it to 3.11, and moving to the 3.5× larger backbone, Iris-Z-big, to 2.92, with fewer unique frames (about 3.6M). With 10 passes the same three runs score 3.99, 3.27 and 3.13 (Iris-Z-big at an earlier checkpoint). The larger backbone is better on both tables. Nothing about depth changed between these runs.

Cost. Tab. 2 sets accuracy against cost. One step of Iris-Z-big takes 0.35 s at 512^2 and 1.4 s at 1024^2 on one H200, so one prediction (40 passes) takes about 15 s at 512^2; Iris-Z-small takes 0.10 s per step and 4.8 s per prediction. A specialist is still much cheaper per image: MoGe-3 takes 0.41 s at 512^2 on the same GPU (0.06 s of it for its 3 refinement steps), about 37 times less. We plan to cut inference time with distillation.

TAB. 2 Accuracy against cost. Accuracy: mean AbsRel×100 over each benchmark table (lower is better). Time: one prediction per image at 512^2 on one H200.

	MEAN ABSREL		
MODEL	MARIGOLD'S SETS	MOGE-3'S SETS	TIME
Iris-Z-big	**2.92**	**3.56**	15 s
Iris-Z-small	3.11	3.84	4.8 s
MoGe-3	3.12	3.70	0.4 s

TAB. 3 MoGe-3's protocol: relative-error L1 alignment, AbsRel×100 (lower is better), on seven of its nine sets (Sec. 5). Reported: as published in MoGe-3's paper [11] (Table C.1, affine-invariant depth). Our runs: run and scored by our code on the same pixels. The ten models with the lowest mean, plus the reference model as its own paper reports it. Shaded bold: best per column.

MODEL	NYUV2	KITTI	ETH3D	IBIMS-1	SINTEL	HAMMER	DIODE	MEAN
Reported								
Depth Pro	3.67	5.12	4.97	3.23	15.8	3.30	4.66	5.82
InfiniDepth	3.97	5.13	4.62	3.13	14.4	3.77	4.94	5.71
Pixel-Perfect Depth	3.29	4.66	3.27	2.94	14.1	2.99	3.95	5.03
UniDepth V2	2.96	3.85	2.94	2.64	13.4	2.48	4.04	4.62
UniK3D	2.81	3.81	2.98	2.54	13.6	2.47	3.57	4.54
MoGe-2	2.88	3.74	2.80	2.37	13.4	2.72	3.12	4.43
MoGe-3 (ViT-G)	2.62	3.42	2.03	2.08	10.9	2.34	**2.65**	3.72
Our runs								
Marigold V2	3.34	4.96	4.83	2.81	14.88	2.80	4.18	5.40
MoGe-3	2.63	3.41	2.01	2.07	10.85	2.30	**2.65**	3.70
Iris-Z-small	2.40	3.53	2.18	2.20	11.50	2.22	2.88	3.84
Iris-Z-big	**2.28**	**2.98**	**1.87**	**1.99**	**10.67**	**1.93**	3.18	**3.56**

TAB. 4 Marigold's protocol: RANSAC alignment in log depth, AbsRel×100 (lower is better). Reported: as published in Marigold V2's paper [16] (Table 1, one decimal; † π^3 trains on ScanNet, not zero-shot). Our runs: run and scored by our code on the same pixels. The ten models with the lowest mean, plus the reference model as its own paper reports it. Shaded bold: best per column.

MODEL	NYUV2	ETH3D	SCANNET	KITTI	DIODE	MEAN
Reported						
InfiniDepth	4.3	6.1	4.7	8.7	6.4	6.04
MoGe-2	3.1	5.0	3.1	6.9	9.6	5.54
PPD (1024)	4.1	4.3	4.6	7.0	6.8	5.36
Lotus-2	3.7	4.1	4.0	6.7	6.5	5.00
FE2E	3.8	3.8	4.3	6.5	5.6	4.80
Marigold V2	3.6	2.8	3.7	5.4	5.2	4.14
π^3	2.9	2.9	2.1†	5.9	4.4	3.64†
Our runs						
Marigold V2	3.58	2.78	3.65	5.41	5.07	4.10
MoGe-3	2.65	2.06	2.86	**3.98**	**4.03**	3.12
Iris-Z-small	2.07	2.12	2.56	4.58	4.22	3.11
Iris-Z-big	**1.90**	**1.76**	**2.25**	4.16	4.50	**2.92**

07 · Discussion

Towards one model. We did not set out to build a better depth estimator. We wanted to know whether a model that only knew RGB could learn the geometry of what it paints, without giving anything up. It could. With no depth head and no new architecture, only one extra embedding, Iris-Z-big now beats MoGe-3, a model built for this one job, on the resolutions we trained on, and it still preserves its generation capabilities. Video and image generators are showing the same thing on other tasks [5, 6], and we think this points to one model that learns every modality of a scene together, and keeps improving with data and compute after task-specific designs have stopped. Depth is the first modality we added; normals, segmentation and multi-view geometry will come next.

Limitations. There are several limitations to this approach. *Depth is still relative.* The prediction is defined up to a scale and shift in log space, so Iris-Z gives neither metric distances nor the camera; MoGe-3 predicts both. This is the gap we want to close next: we will add one more group of tokens to the same sequence that predicts the camera intrinsics. With the camera known, the depth map becomes a point cloud from which the shift can be obtained, so what remains should be a single global scale. *Resolution and depth range.* We train at up to about 1024^2 pixels and do not evaluate on 2 to 8 MP images (Spring, Synth4K). *Cost.* Each prediction takes 40 passes of a 9B or 32B transformer: about 15 s per 512^2 image for Iris-Z-big against MoGe-3's 0.41 s on the same GPU. We made that trade on purpose. The open question was whether one generative model could match the specialists at all; speed is a problem we already know how to solve the same way image generation did. We believe this is the right path, and that the optimisations will follow. *Generation drift.* Generated images move towards the look of the training data: the human preference score of Iris-Z-big falls from 26.71 to 24.30, prompt adherence drops slightly (CLIP 27.42 against 27.85) and photorealism rises. *Single images.* We evaluate one image at a time; consistency across video frames or views is not measured.

References

[01] T. B. Brown et al. Language Models are Few-Shot Learners. In NeurIPS, 2020.

[02] J. Kaplan et al. Scaling Laws for Neural Language Models. arXiv:2001.08361, 2020.

[03] J. Hoffmann et al. Training Compute-Optimal Large Language Models. arXiv:2203.15556, 2022.

[04] R. Sutton. The Bitter Lesson. `http://www.incompleteideas.net/IncIdeas/BitterLesson.html`, 2019.

[05] T. Wiedemer, Y. Li, P. Vicol, S. S. Gu, N. Matarese, K. Swersky et al. Video Models are Zero-Shot Learners and Reasoners. arXiv:2509.20328, 2025.

[06] V. Gabeur et al. Image Generators are Generalist Vision Learners. arXiv:2604.20329, 2026.

[07] C. Deng et al. Emerging Properties in Unified Multimodal Pretraining. arXiv:2505.14683, 2025.

[08] C. Wei et al. PixelUMM: Encoder-Free Unified Image and Video Understanding and Generation. arXiv:2609.38597, 2026.

[09] R. Wang, S. Xu, C. Dai, J. Xiang et al. MoGe: Unlocking Accurate Monocular Geometry Estimation for Open-Domain Images with Optimal Training Supervision. In CVPR, 2025.

[10] R. Wang et al. MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details. arXiv:2507.02546, 2025.

[11] L. Kong, R. Li, R. Wang, S. Xu, C. Yao, J. Xiang et al. MoGe-3: Fine-Detail Monocular Geometry Estimation with Self-Guided Sparse Volumetric Refinement. arXiv:2607.17967, 2026.

[12] A. Bochkovskii et al. Depth Pro: Sharp Monocular Metric Depth in Less Than a Second. arXiv:2410.02073, 2024.

[13] H. Lin et al. Depth Anything 3: Recovering the Visual Space from Any Views. arXiv:2511.10647, 2025.

[14] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, K. Schindler. Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation. In CVPR, 2024.

[15] G. Martin Garcia et al. Fine-Tuning Image-Conditional Diffusion Models is Easier than You Think. In WACV, 2025.

[16] I. Pavlovic, T. Wandel, A. Obukhov, L. Bartolomei, A. Davydov, F. Tosi, M. Poggi, S. Süsstrunk, D. Dai. Marigold V2: Revisiting Diffusion Transformers for Monocular Depth Estimation. arXiv:2609.08084, 2026.

[17] H. Li Cai, C. Garabito. Iris-3B: Going Beyond the Latent with Pixel-Space Diffusion Training, Conversion and Fine-Tuning. arXiv:2610.09450, 2026.

[18] Black Forest Labs. FLUX.2 [dev], model card on Hugging Face, 2025.

[19] Black Forest Labs. FLUX.2 [klein] Base 9B, model card on Hugging Face, 2025.

[20] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, M. Le. Flow Matching for Generative Modeling. In ICLR, 2023.

[21] A. Radford et al. Learning Transferable Visual Models From Natural Language Supervision. In ICML, 2021.

[22] X. Wu et al. Human Preference Score v2: A Solid Benchmark for Evaluating Human Preferences of Text-to-Image Synthesis. arXiv:2306.09341, 2023.

[23] D. Eigen, C. Puhrsch, R. Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In NeurIPS, 2014.

[24] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, V. Koltun. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. IEEE TPAMI, 2022. arXiv:1907.01341.

[25] R. Ranftl, A. Bochkovskiy, V. Koltun. Vision Transformers for Dense Prediction. In ICCV, 2021.

[26] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, M. Müller. ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth. arXiv:2302.12288, 2023.

[27] W. Yin et al. Metric3D: Towards Zero-shot Metric 3D Prediction from A Single Image. In ICCV, 2023.

[28] L. Piccinelli et al. UniDepth: Universal Monocular Metric Depth Estimation. In CVPR, 2024.

[29] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, H. Zhao. Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. In CVPR, 2024.

[30] L. Yang et al. Depth Anything V2. In NeurIPS, 2024.

[31] X. Fu et al. GeoWizard: Unleashing the Diffusion Priors for 3D Geometry Estimation from a Single Image. In ECCV, 2024.

[32] M. Gui et al. DepthFM: Fast Monocular Depth Estimation with Flow Matching. arXiv:2403.13788, 2024.

[33] J. He et al. Lotus: Diffusion-based Visual Foundation Model for High-quality Dense Prediction. arXiv:2409.18124, 2024.

[34] W. Hu et al. DepthCrafter: Generating Consistent Long Depth Sequences for Open-world Videos. arXiv:2409.02095, 2024.

[35] G. Ben Melech Stan et al. LDM3D: Latent Diffusion Model for 3D. arXiv:2305.10853, 2023.

[36] J. Zhang et al. JointNet: Extending Text-to-Image Diffusion for Dense Distribution Modeling. In ICLR, 2024.

[37] X. Long et al. Wonder3D: Single Image to 3D using Cross-Domain Diffusion. In CVPR, 2024.

[38] Aether Team, H. Zhu et al. Aether: Geometric-Aware Unified World Modeling. arXiv:2503.18945, 2025.

[39] A. Bar, Y. Gandelsman, T. Darrell, A. Globerson, A. A. Efros. Visual Prompting via Image Inpainting. In NeurIPS, 2022.

[40] X. Wang, W. Wang, Y. Cao, C. Shen, T. Huang. Images Speak in Images: A Generalist Painter for In-Context Visual Learning. In CVPR, 2023.

[41] Y. Gan et al. InstructCV: Instruction-Tuned Text-to-Image Diffusion Models as Vision Generalists. In ICLR, 2024.

[42] S. Xiao et al. OmniGen: Unified Image Generation. arXiv:2409.11340, 2024.

[43] C. Zhou et al. Transfusion: Predict the Next Token and Diffuse Images with One Multi-Modal Model. arXiv:2408.11039, 2024.

[44] C. Wu et al. Janus: Decoupling Visual Encoding for Unified Multimodal Understanding and Generation. arXiv:2410.13848, 2024.

[45] W. Peebles, S. Xie. Scalable Diffusion Models with Transformers. In ICCV, 2023.

[46] P. Esser et al. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In ICML, 2024.

A · Examples

All examples are from Iris-Z-big, chosen from the held-out prompts and benchmark images of our evaluation.

A.1 Image → depth

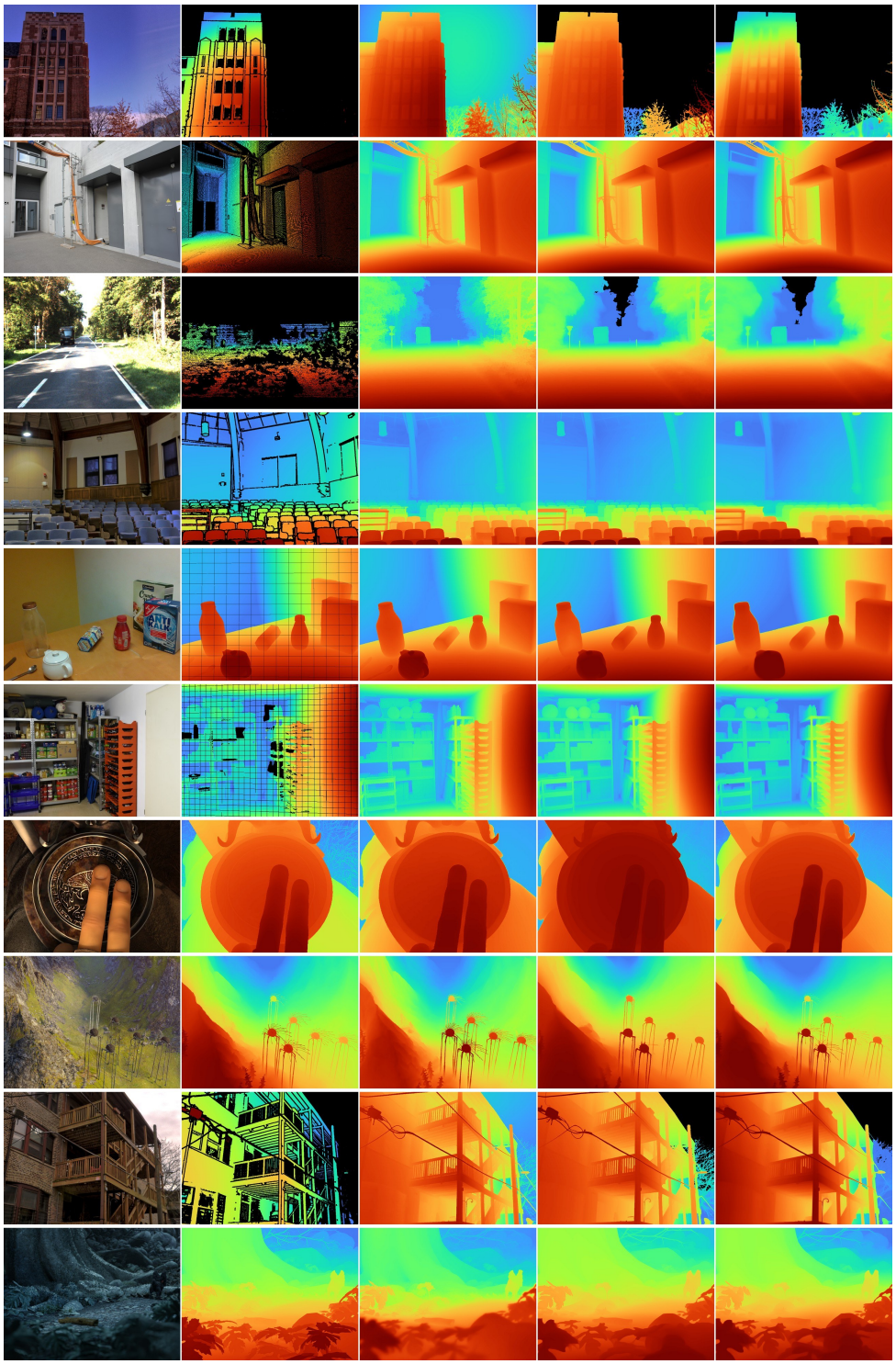


FIG. 6 Benchmark photographs with their ground truth (black: no measurement) and each model's prediction. Each prediction is aligned as its table scores it and coloured on the ground truth's range (near = red, far = blue). Black: no depth or sky; ours and MoGe-3 mark sky with their own sky masks, and Marigold V2 has no sky output.

A.2 Prompt → RGBD

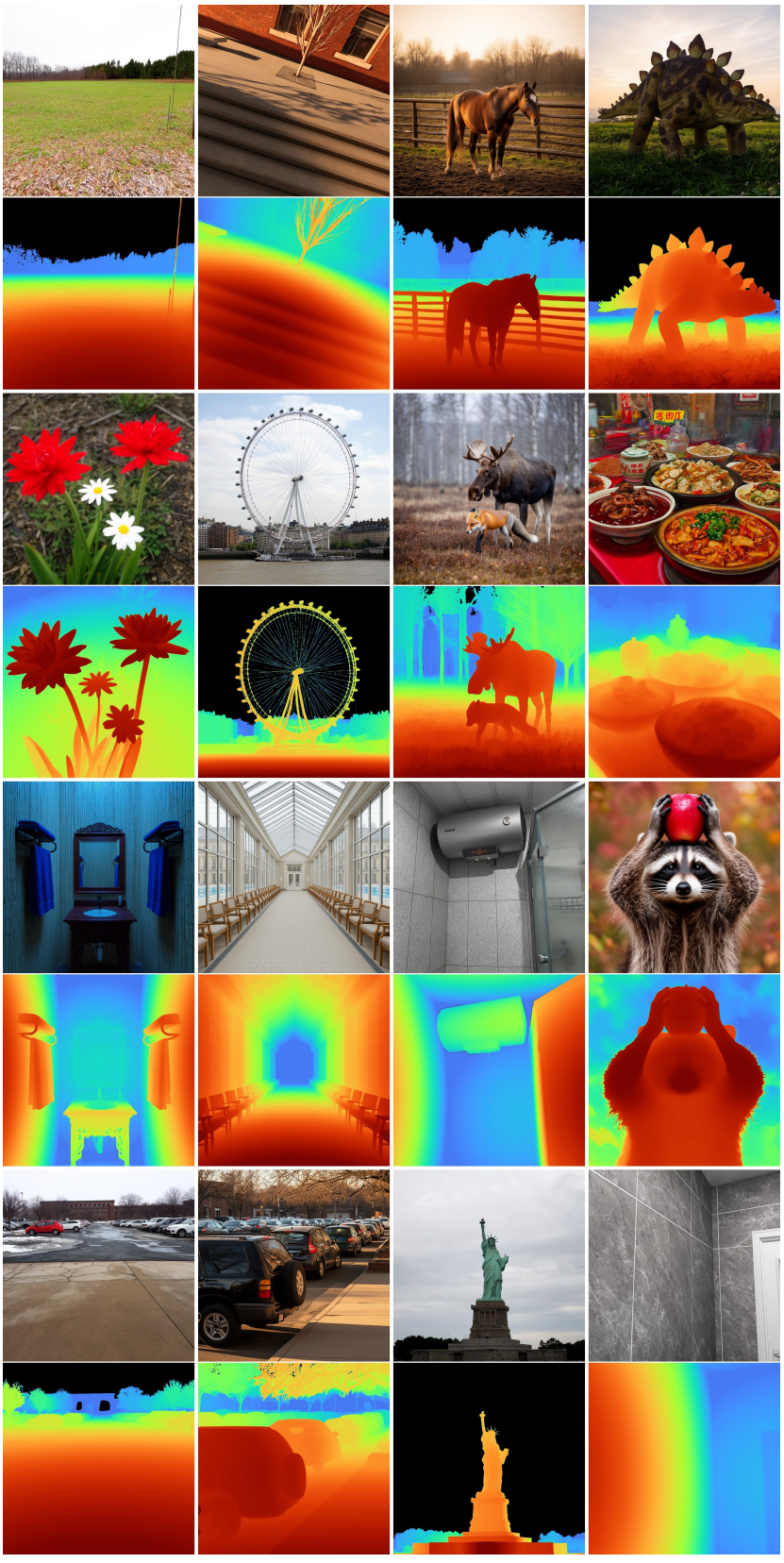


FIG. 7 Prompt → image and depth, generated together from held-out test prompts; each image above its depth.

A.3 Depth → image

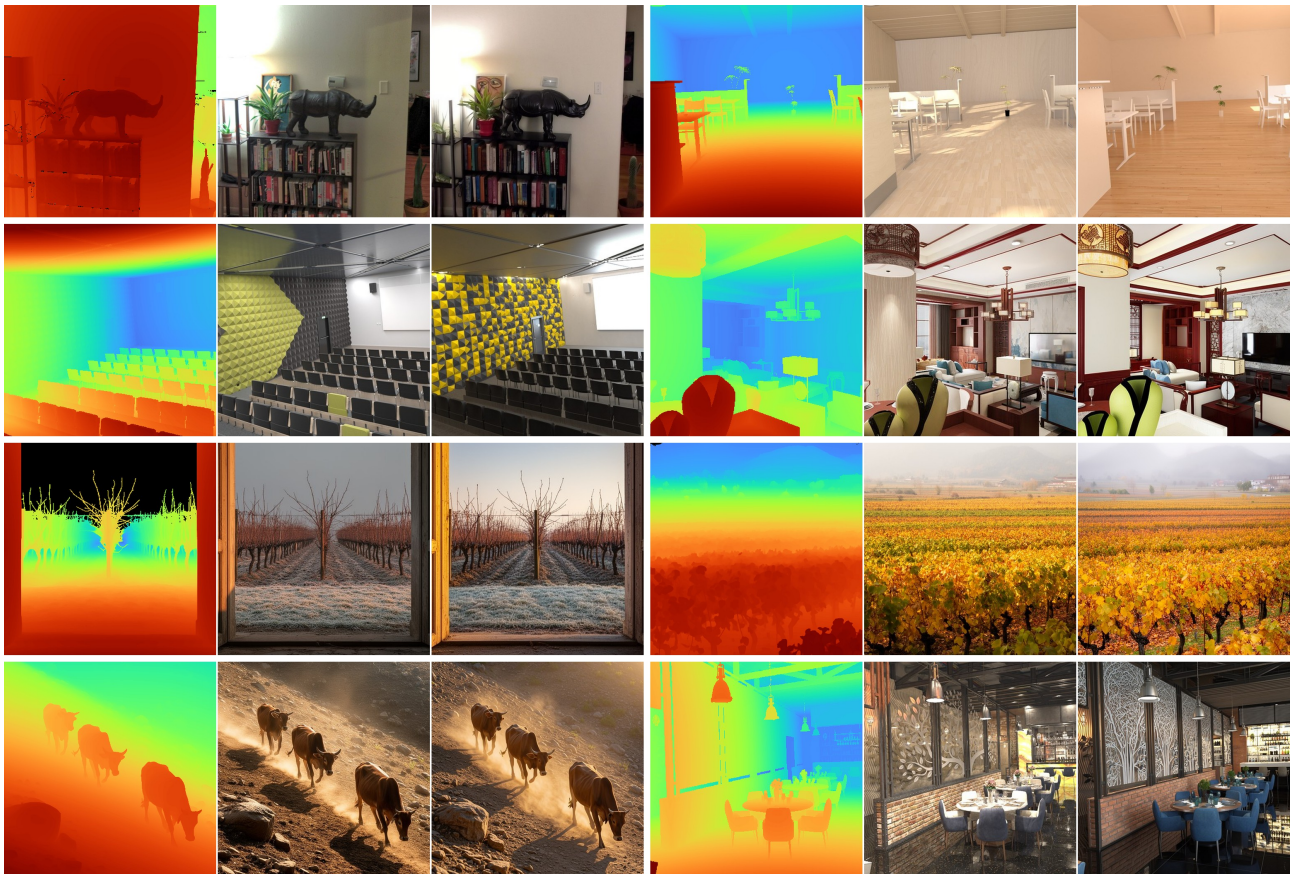


FIG. 8 Depth → image on held-out frames: the given depth, the real image, and Iris-Z-big's image painted onto the depth with the frame's caption.

A.4 Base model vs. ours

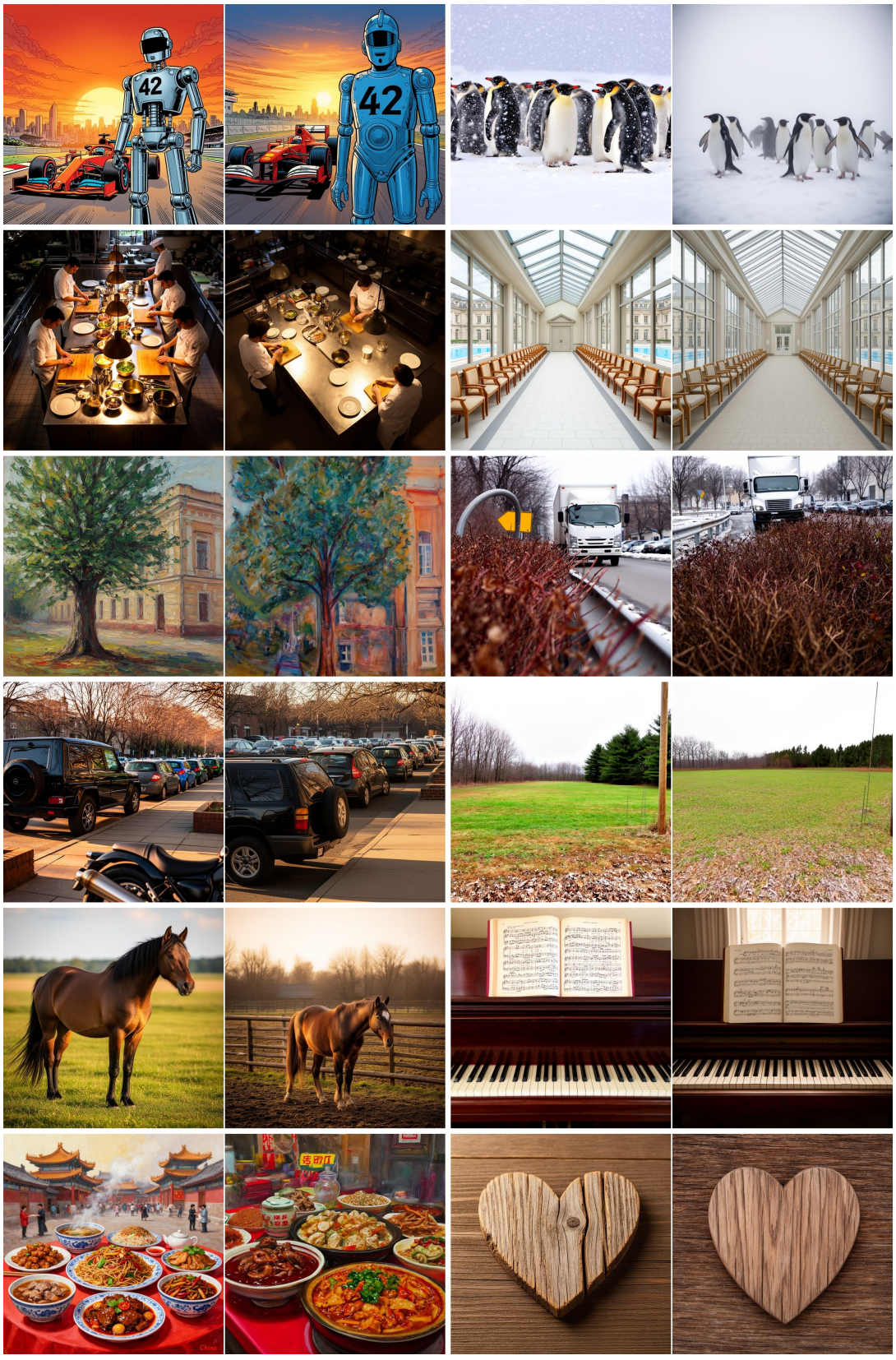


FIG. 9 Same prompt and seed, left the untouched base model, right Iris-Z-big (the image from a joint RGBD generation).

B · Depth accuracy: δ_1

The same runs as Tab. 3 and Tab. 4, scored with δ_1: the share of valid pixels whose aligned prediction is within a factor of 1.25 of the ground truth, in % (higher is better). Same alignments, sets and resolutions as the AbsRel tables (Sec. 5).

TAB. 5 MoGe-3's protocol on seven of its nine sets, δ_1 in % (higher is better). Reported: as published in MoGe-3's paper [11] (Table C.1, affine-invariant depth). Our runs: run and scored by our code on the same pixels. Same models as Tab. 3. Shaded bold: best per column.

MODEL	NYUV2	KITTI	ETH3D	IBIMS-1	SINTEL	HAMMER	DIODE	MEAN
Reported								
Depth Pro	98.2	96.8	96.4	98.3	80.1	**99.6**	95.6	95.00
InfiniDepth	97.7	96.2	96.5	98.5	81.8	99.1	95.2	95.00
Pixel-Perfect Depth	98.4	97.3	98.6	98.6	82.0	99.2	96.6	95.81
UniDepth V2	98.6	98.1	98.5	98.4	83.2	**99.6**	96.5	96.13
UniK3D	98.6	98.1	98.6	98.4	83.2	**99.6**	97.0	96.21
MoGe-2	98.6	98.1	99.1	98.9	82.3	99.5	97.6	96.30
MoGe-3 (ViT-G)	98.7	98.4	99.5	98.9	**86.9**	**99.6**	98.0	97.14
Our runs								
Marigold V2	98.44	96.64	97.71	98.49	80.01	99.46	95.82	95.22
MoGe-3	98.66	98.40	**99.54**	98.86	86.86	99.58	**98.13**	**97.15**
Iris-Z-small	98.76	98.48	99.13	98.80	85.30	99.59	97.59	96.81
Iris-Z-big	**98.85**	**98.85**	99.34	**99.10**	86.64	99.38	97.05	97.03

TAB. 6 Marigold's protocol, δ_1 in % (higher is better). Reported: as published in Marigold V2's paper [16] (Table 1, one decimal; † π^3 trains on ScanNet, not zero-shot). Our runs: run and scored by our code on the same pixels. Same models as Tab. 4. Shaded bold: best per column.

MODEL	NYUV2	ETH3D	SCANNET	KITTI	DIODE	MEAN
Reported						
InfiniDepth	97.6	95.4	96.9	92.3	95.8	95.60
MoGe-2	98.4	96.2	98.3	89.9	84.5	93.46
PPD (1024)	97.7	98.0	97.2	95.5	95.9	96.86
Lotus-2	97.6	98.6	97.2	94.1	95.5	96.60
FE2E	97.6	98.7	97.1	96.0	96.4	97.16
Marigold V2	98.0	99.2	97.9	97.4	97.1	97.92
π^3	98.7	98.9	99.1†	96.7	96.8	98.04†
Our runs						
Marigold V2	98.01	99.15	97.89	97.42	97.05	97.90
MoGe-3	98.49	**99.62**	98.31	98.30	97.57	98.46
Iris-Z-small	99.25	99.53	98.71	98.04	**98.00**	98.71
Iris-Z-big	**99.28**	99.56	**98.85**	**98.59**	97.65	**98.78**