Marigold: A machine learning-based web app for zebrafish pose tracking

Gregory Teicher[1,2], R. Madison Riffe[1,3], Wayne Barnaby[1,3], Gabrielle Martin[1], Benjamin E. Clayton[1], Josef G. Trapani[3,4,5], Gerald B. Downes[1,2,3*]

[1] Biology Department, University of Massachusetts Amherst, Amherst, MA, USA
[2] Molecular and Cellular Biology Graduate Program, University of Massachusetts Amherst, Amherst, MA, USA
[3] Neuroscience and Behavior Graduate Program, University of Massachusetts Amherst, Amherst, MA, USA
[4] Biology Department, Amherst College, Amherst, MA, USA
[5] Neuroscience Program, Amherst College, Amherst, MA, USA
* Correspondence: gbdownes@umass.edu

Abstract

High-throughput behavioral analysis is important for drug discovery, toxicological studies, and modeling neurological disorders such as autism and epilepsy. Zebrafish embryos and larvae are ideal for such applications because they are spawned in large clutches, develop rapidly, feature a relatively simple nervous system, and have orthologs to many human disease genes. However, existing software for video-based behavioral analysis can be incompatible with recordings that contain dynamic backgrounds or foreign objects, lack support for multiwell formats, require expensive hardware, and/or demand considerable programming expertise. Here, we introduce Marigold, a free and open source web app for high-throughput behavioral analysis of embryonic and larval zebrafish. Marigold features an intuitive graphical user interface (GUI), tracks up to 10 user-defined keypoints, supports both single- and multiwell formats, and exports a range of kinematic parameters in addition to publication-quality data visualizations. By leveraging a highly efficient, custom-designed neural network architecture, Marigold achieves competitive training and inference speeds even on modestly powered computers lacking a discrete graphics processing unit (GPU). As a web app, Marigold does not require any installation and runs within popular web browsers on ChromeOS, Linux, macOS, and Windows. To demonstrate Marigold's utility, we used two sets of experiments. First, we examined novel aspects of the touch-evoked escape response in *techno trousers (tnt)* mutant embryos, which contain a previously described loss-of-function mutation in the gene encoding Eaat2b, a glial glutamate transporter. We identified differences and interactions between touch location (head vs. tail) and genotype. Second, we investigated the effects of feeding on larval visuomotor behavior at 5 and 7 days post-fertilization (dpf). We found differences in the number and vigor of swimming bouts between fed and unfed fish at both time points, as well as interactions between developmental stage and feeding regimen. In both biological experiments presented here, the use of Marigold facilitated novel behavioral findings. Marigold's ease of use, robust pose tracking, amenability to diverse experimental paradigms, and flexibility regarding hardware requirements make it a powerful tool for analyzing zebrafish behavior, especially in low-resource settings such as course-based undergraduate research experiences (CUREs). Marigold is available at: https://downeslab.github.io/marigold/.

Keywords: zebrafish, behavior, pose tracking, machine learning, web app, software, open source

Introduction

The kinematic and ethological analysis of animal movements is an essential task for many researchers in neuroscience, psychology, and related fields. For investigators working with zebrafish (*Danio rerio*), behavioral assays are key methods for characterizing the toxicological effects of environmental pollutants, identifying therapeutic agents using small molecule screens, and modeling neurological disorders such as autism and epilepsy (Baraban et al., 2013; d'Amora & Giordani, 2018; MacRae & Peterson, 2023; Sakai et al., 2018). Zebrafish embryos are spawned in large clutches and develop rapidly, making high-throughput screening in multiwell plates an attractive experimental paradigm. Additionally, zebrafish have a well characterized behavioral repertoire, a relatively simple nervous system containing fewer neurons compared to many other vertebrates, and orthologs to numerous human disease genes (Berg et al., 2018; Holtzman et al., 2016; Howe et al., 2013; Kalueff et al., 2013). Thus, zebrafish are an excellent system for studying vertebrate behavior.

Despite the rising popularity of behavioral screening in zebrafish, currently available software for analysis has a variety of limitations. Behavioral analysis software has traditionally utilized image processing techniques such as pixel intensity thresholding, background subtraction, skeletonization, and center of mass calculations to isolate animals from the visual background and track their movements (Burgess & Granato, 2007; Clift et al., 2014; Joo et al., 2020; Marques et al., 2018; Mirat et al., 2013). These techniques are often sufficient, but can lack robustness when challenged with dynamic backgrounds, varying lighting conditions, or earlier developmental stages when embryos present with less contrast against the background. Many solutions provide relatively limited kinematic information by only tracking a single point representing a larva's overall position, while those programs capable of tracking multiple points often fail when foreign objects enter the field of view. This type of approach is particularly problematic for touch-evoked response assays that require a probe to enter the field of view, especially when the tactile stimulus is applied to the tail rather than the more easily detectable head. Additionally, many programs are closed source and/or tied to proprietary hardware such as specialized imaging cabinets, thus restricting customizability.

Several recently introduced programs have extended machine learning-based advances in human pose tracking to other species (Gore et al., 2023; Graving et al., 2019; Mathis et al., 2018; Nilsson et al., 2020; Pereira et al., 2019; Ravan et al., 2023; Romero-Ferrero et al., 2019). While the new pose tracking programs can exhibit high accuracy, they can also demand substantial programming or other technical expertise—not to mention a high-end graphics processing unit (GPU)—to train and operate the large deep learning models. Additionally, many of these programs require workflows combining GUI-, command line-, and cloud-based elements. These factors can make such software difficult to use for those who lack programming or machine learning experience. It also limits the use of such programs in resource-constrained settings and course-based undergraduate research

experiences (CUREs), where such software might otherwise be useful. Lastly, due to the general-purpose nature of existing deep learning-based pose tracking programs, they often lack support for popular zebrafish experimental paradigms, such as imaging in multiwell plates, which limits their utility for zebrafish researchers.

Here, we present Marigold, a machine learning program for automated pose tracking of embryonic and larval zebrafish that unites many of the strengths of existing programs, while also addressing several common limitations. Marigold reliably tracks up to 10 user-defined body keypoints, supports both single- and multi-well configurations, and features a user-friendly interface (Figure 1). Marigold facilitates a streamlined workflow consisting of three stages: (1) generating a dataset by extracting a modest number of frames from representative behavioral recordings and labeling the frames with animal pose coordinates; (2) training a neural network using the generated dataset; (3) analyzing new behavioral recordings using the trained neural network and exporting the resulting raw kinematic parameters and visualizations. Despite using convolutional neural networks, which are notoriously computationally demanding, Marigold is able to train models and analyze behavioral recordings at competitive speeds on current desktop or laptop computers without requiring a dedicated GPU. This speed is achieved by using a highly efficient neural network architecture incorporating a series of macro-architectural and micro-architectural optimizations. For each animal analyzed, Marigold generates publication-ready trajectory plots and a CSV file of keypoint coordinates, velocities, and angles between keypoints, which facilitates downstream analysis pipelines. As a web app, Marigold does not require any installation and is highly cross-platform, running within popular web browsers on ChromeOS, Linux, macOS, and Windows. Marigold is free and open source software.

To showcase its robustness and versatility, we used Marigold to obtain biological insights from two sets of zebrafish experiments. First, we examined novel aspects of the embryonic touch-evoked response phenotype in the previously described *techno trousers (tnt)* mutant (Granato et al., 1996; McKeown et al., 2012). Second, we investigated the effects of feeding on larval visuomotor response behavior. Larvae begin hunting for food by 5 dpf yet can survive solely on nutrients from their yolks until at least 7 dpf (Hernandez et al., 2018; Lucore & Connaughton, 2021; Schwartz et al., 2021), making 5–7 dpf a popular development window for experimentation. While it is known that developmental stage can influence the larval visuomotor response (de Esch et al., 2012; Y. Liu et al., 2015), the impact of feeding and its interaction with developmental stage have not been examined. Together, these experiments exemplify the types of biological questions that can be addressed using the detailed kinematic analysis provided by Marigold.

Methods

Zebrafish husbandry. All animals used in this study were obtained from our laboratory's fish facility. Adult zebrafish were maintained on a 14 h light, 10 h dark cycle at approxi-

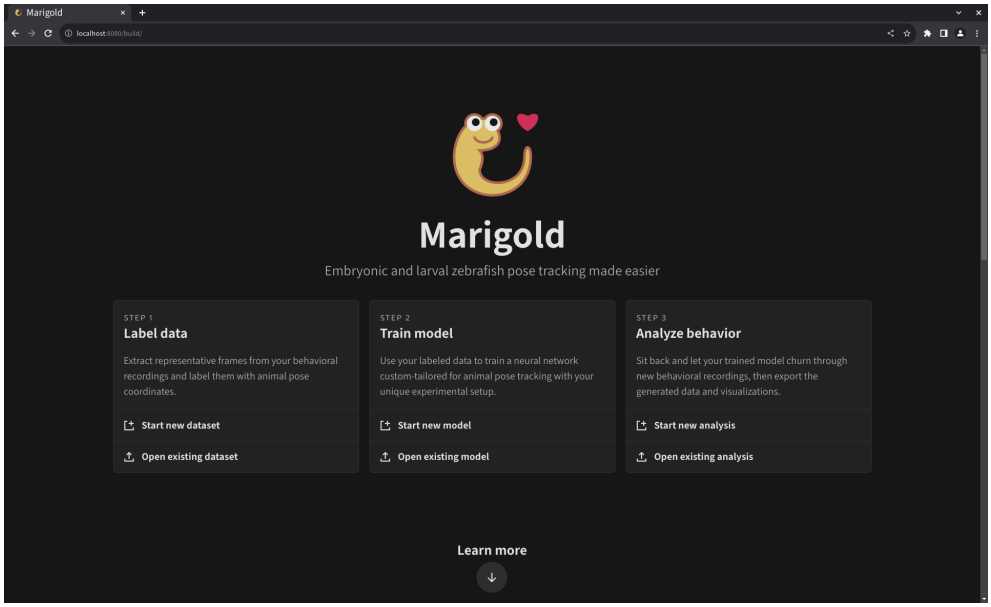

Figure 1. Marigold features an intuitive web-based graphical user interface (GUI). Marigold's welcome screen (as viewed in Google Chrome) orienting the user to the three steps of using the program: creating a dataset, training a model, and analyzing behavior.

mately 28.5 °C. Adult zebrafish were bred to obtain larvae. Larvae younger than 5 dpf were maintained in the dark at 28.5 °C. *techno trousers* (*slc1a2b*^{tk57}) fish (Granato et al., 1996; McKeown et al., 2012) were maintained on a Tüpfel long-fin (TL) background. All wild type fish were from either Tübingen or TL strains as indicated. All animal procedures were approved by the University of Massachusetts Amherst Institutional Animal Care and Use Committee (IACUC) under Animal Welfare Assurance number A3551-01. No animal procedures were performed at Amherst College.

Training data collection. Two novel datasets were developed: one for touch-evoked responses and one for visuomotor responses. For both datasets, embryos were obtained from mass matings of Tübingen or TL fish. Behavior was monitored using an IDT X-Stream 1440p PCIe 2.0 high speed camera (Integrated Design Tools, Inc., Pasadena, CA, USA). Larvae were imaged from above and illuminated using a white light or near-infrared source from below. Recordings were made under varying conditions of lighting intensity, resolution, magnification, and focus.

From the generated movies, frames were manually selected for inclusion in the dataset with the goal of maximizing the representation of diverse morphologies and poses from within each movie. Each frame was then manually labeled and the labeled coordinates were used to generate target heatmaps (i.e., two-dimensional Gaussians) that served as the "ground truth" for our neural networks during training.

For the touch-evoked response dataset, movies were recorded of larvae between the developmental stages of 2–4 dpf in wells in the lids of 24-well plates. Manual labels

targeted 7 keypoints evenly spaced across the embryo's/larva's rostral caudal axis. For the visuomotor response dataset, movies were recorded of larvae between the developmental stages of 5–7 dpf in 24-well plates. Manual labels targeted a single keypoint positioned between the two eyes.

Neural network design and training. Datasets were randomly divided into training (75%) and validation (25%) splits. Image pixel intensities were shifted from the integral range 0–255 to the floating point range 0.0–1.0. During training, we made use of data augmentation (Shorten & Khoshgoftaar, 2019) by randomly performing vertical and horizontal flips, rotating up to 45 degrees clockwise or counter-clockwise, shifting brightness by up to 10%, and adding Gaussian noise with a mean of 0 and standard deviation up to 0.01. Finally, we subtract the per-image mean and divide by the per-image standard deviation before providing each image as input to the neural networks.

Neural networks were trained at an input resolution of 448×448 (touch-evoked response dataset) or 256×256 (visuomotor response dataset). For the hourglass macro-architecture, intro blocks consisted of a 4×4 convolution stride 2×2 stride followed by a normalization layer. Outro blocks consisted of a normalization layer, a 1×1 convolution layer with a channel expansion ratio of 8, a hard swish layer, and a 1×1 convolution layer reducing the channels to the number of keypoints. For the isometric macro-architecture, intro blocks consisted of a 8×8 convolution with 8×8 stride followed by a normalization layer. Outro blocks consisted of a normalization layer, a 1×1 convolution layer with a channel expansion ratio of 8, a hard swish layer, a 1×1 convolution layer reducing the channels to 16× the number of keypoints, and a pixel shuffle layer transforming the number of channels to the number of keypoints while increasing the spatial resolution. All MobileNetV3 blocks and variations used an expansion ratio of 3, with the number of input channels set to 32 (touch-evoked response dataset) or 16 (visuomotor response dataset) at the smallest spatial resolution.

Training was performed using mean squared error (MSE) loss, batch size equal to 8 (touch-evoked response dataset) or 32 (visuomotor response dataset), and the Adam optimizer (Kingma & Ba, 2017) with learning rate = 0.005, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1 \times 10^{-6}$. Convolution and transposed convolution layer weights and biases were initialized using Xavier uniform initialization (Glorot & Bengio, 2010) and zeros, respectively. We set the ε of normalization layers to 1×10^{-4}.

Performance benchmarks. Performance was measured using a Lenovo Thinkpad P14s Gen 4 laptop equipped with an AMD Ryzen 7 PRO 7840U central processing unit (CPU) and running Fedora Workstation 40 with Linux kernel version 6.8.8. Neural network benchmarks were obtained using Python 3.12.2, pandas 2.2.2, NumPy 1.26.4, SciPy 1.13.0, PyTorch 2.3.0, and torchvision 0.18.0.

Software development. Where indicated, neural network designs were prototyped and benchmarked in Python using the NumPy, pandas, PyTorch, and torchvision packages. However, Marigold itself is written using HTML, CSS, JavaScript, and C++, with the latter being compiled to WebAssembly. WebAssembly output was generated using the LLVM (version 18.1.1) Clang compiler and wasm-ld linker. Marigold does not use any external libraries or components other than the font (Adobe Source Sans 3, used under the SIL Open Font License Version 1.1). Testing and building for deployment were automated using GitHub Actions, with hosting provided by GitHub Pages.

Touch-evoked response assay. Embryos were obtained from mass matings of heterozygous *techno trousers* (*slc1a2b*$^{+/tk57}$) fish. Behavior was monitored using an IDT X-Stream 1440p PCIe 2.0 high speed camera (Integrated Design Tools, Inc., Pasadena, CA, USA) in a temperature controlled setting (25 °C). Larvae were transferred in E3 solution to a well in the inverted lid of a CytoOne 24-Well Tissue Culture Plate (USA Scientific, Inc., Ocala, FL, USA) and allowed to acclimate briefly before touching on either the side of the head or tip of the tail as indicated with a 3.22/0.16 g of force von Frey filament held by a surgical blade holder. Larvae were imaged from above and illuminated using a white light source from below. Responses were recorded at a resolution of 1024×1024 pixels with a framerate of 1000 Hz and exposure time of 0.8 ms. Recordings were manually trimmed to remove extraneous frames before and after the response, as determined by the first and last visible movement initiated by the fish, respectively.

***slc1a2b* genotyping.** Following behavioral analysis, embryos were euthanized using an overdose of MS-222 (pH = 7.0) (Sigma-Aldrich, St. Louis, MO, USA). DNA was extracted from the euthanized embryos using Extract-N-Amp Tissue PCR Kit (Sigma-Aldrich, St. Louis, MO, USA) according to the manufacturer's protocol. A 192 bp region of exon 2 of *slc1a2b* flanking the site encoding the A393V mutation was amplified by PCR using the forward primer `5'-TGCTGGAACTCTGCCCGTGA-3'` and the reverse primer `5'-ACGGTGACGATCTGTCCAGG-3'`. PCR was performed using AmpliTaq Gold DNA Polymerase (Applied Biosystems/Thermo Fisher Scientific, Waltham, MA, USA) according to the manufacturer's protocol and using an annealing temperature of 59 °C. The PCR product was digested overnight using Fnu4HI (New England Biolabs, Ipswich, MA, USA) according to the manufacturer's protocol. Digestion yielded a single fragment of 192 bp for homozygous mutant fish (*slc1a2b*$^{tk57/tk57}$), two fragments of 133 and 59 bp for homozygous wild type fish (*slc1a2b*$^{+/+}$), and three fragments of 192, 133, and 59 bp for heterozygous fish (*slc1a2b*$^{+/tk57}$). The digested PCR products were resolved for genotypic determination by agarose gel electrophoresis.

Visuomotor response assay. Embryos were obtained from mass matings of Tübingen strain zebrafish and maintained in E3 media for the duration of experiments. Larvae

were kept in petri dishes until 5 dpf, at which point they were transferred to 2.5 L fish tanks at a density of approximately 45 fish/tank and maintained on a 14 h light, 10 h dark light cycle. The initial volume of E3 media in the tanks was 250 mL, with an additional 250 mL added daily. Fish in fed conditions were given approximately 50 mg of Gemma Micro 75 (Skretting, Stavanger, Norway) once daily beginning at 5 dpf and continuing through 7 dpf. Feeding occurred at approximately 12:00 PM each day, with behavioral recordings occurring between 4-6 h after feeding. Behavior was monitored using an IDT X-Stream 1440p PCIe 2.0 high speed camera (Integrated Design Tools, Inc., Pasadena, CA, USA) in a temperature controlled setting (25 °C). Larvae were transferred to the wells of a CytoOne 24-Well Tissue Culture Plate (USA Scientific, Inc., Ocala, FL, USA) for recording. Larvae were imaged from above and illuminated from below using a custom-built light source which was toggled between illumination modes using either near-infrared light or near-infrared light plus white light. Fish were shielded from ambient light by an enclosing cabinet. Responses were recorded at a resolution of 1440 x 1440 pixels with a framerate of 20 Hz and exposure time of 0.8 ms.

Statistical analysis and data visualization. Statistical significance was determined by two-way analysis of variance (ANOVA) with Tukey post hoc test using the R language (version 4.3.3) (R Core Team, 2024). Except where otherwise noted, data visualizations were generated using the ggplot2 library (version 3.5.1) for R (Wickham, 2016).

Results

Neural network design. Artificial neural networks are universal function approximators that are currently among the most popular machine learning techniques for computer vision tasks (Greener et al., 2022). Modern neural network architectures are typically composed of many layers arranged in a nonlinear order, with each layer responsible for performing a characteristic mathematical operation on its input to produce an output. While some layers perform fixed operations, other layers (e.g., convolution layers) have parameters that are initialized to small random values and continuously updated during training. The training process is typically implemented using backpropagation, which allows the parameters to be updated according to the error signal generated by comparing the network's output to a human-labeled "ground truth" output according to some loss metric. In this way, the network learns a model that approximates the relationship between input-output pairs in a partition of a dataset called the "training set". By evaluating the model's loss on a holdout partition of the input-output pairs called the "validation set", one can estimate the model's ability to generalize to previously unseen data.

Arguably, modern machine learning for computer vision tasks is dominated by the massively parallel training of overparameterized neural networks on colossal datasets using high-end GPU clusters. This paradigm is at odds with the use case scenarios in academic biology labs and course-based undergraduate research experiences (CUREs).

In these settings, available hardware often lags behind the cutting edge and training data is scarce and typically must be collected and labeled *de novo*.

Motivated by these constraints, we sought to design a neural network architecture with the following properties. First, the architecture should facilitate rapid training, even with small datasets. Second, the architecture, once trained, should be capable of fast inference. Third, the architecture should be straightforward to tailor for a variety of experimental paradigms. Fourth, the architecture should lend itself to a performant implementation even when running within a web app on modest hardware.

In order to evaluate possible solutions to these specifications, we collected and labeled two novel datasets and prototyped neural network architectures using PyTorch, a highly optimized machine learning library for Python. Our datasets consisted of frames from touch-evoked response movies in the lids of 24-well plates, which we labeled with 7 keypoints evenly spaced across the rostral–caudal axis, or frames from visuomotor response movies in the wells of 24-well plates, which we labeled with a single keypoint located between the two eyes (Figure 2).

We began our design process by evaluating two potential macro-architectural approaches (Figure 3A). In the first approach, we adopted an hourglass-shaped architecture. In this architecture, as information flows through the successive layers of the network, it is first downsampled, then upsampled. The downsampling or "encoding" path helps to minimize the computational cost of the expensive convolutional layers and pack the information into a compact representation by passing it through a representational bottleneck, while also helping to aggregate spatial context and provide some degree of translational and scale invariance. The upsampling or "decoding" path returns the transformed information to half the full spatial resolution, which is required in order for the network to produce probability heatmaps for each body keypoint of interest. Long-range "skip" connections help the network to recover spatial information that may be lost during downsampling. Variations on this design have demonstrated widespread success in tasks such as human pose estimation and medical image segmentation (Newell et al., 2016; Ronneberger et al., 2015).

In the second approach, we adopted an isometric architecture (Sandler, Baccash, et al., 2019). This architecture is characterized by its constant internal resolution throughout most of its layers. This attribute is in stark contrast to most architectures used for dense prediction tasks such as pose estimation and image segmentation, which typically feature some form of progressive down- and upsampling along with merging of information from multiple internal resolutions. Compared to hourglass-shaped architectures, isometric architectures remain relatively unexplored. Interestingly, an isometric architecture was recently used for keypoint-based fish morphometric analysis (Saleh et al., 2023), suggesting that it could have potential for pose estimation. To facilitate comparison between the two macro-architectural approaches, we construct both implementations around the MobileNetV3 inverted bottleneck residual block, which was designed for fast inference

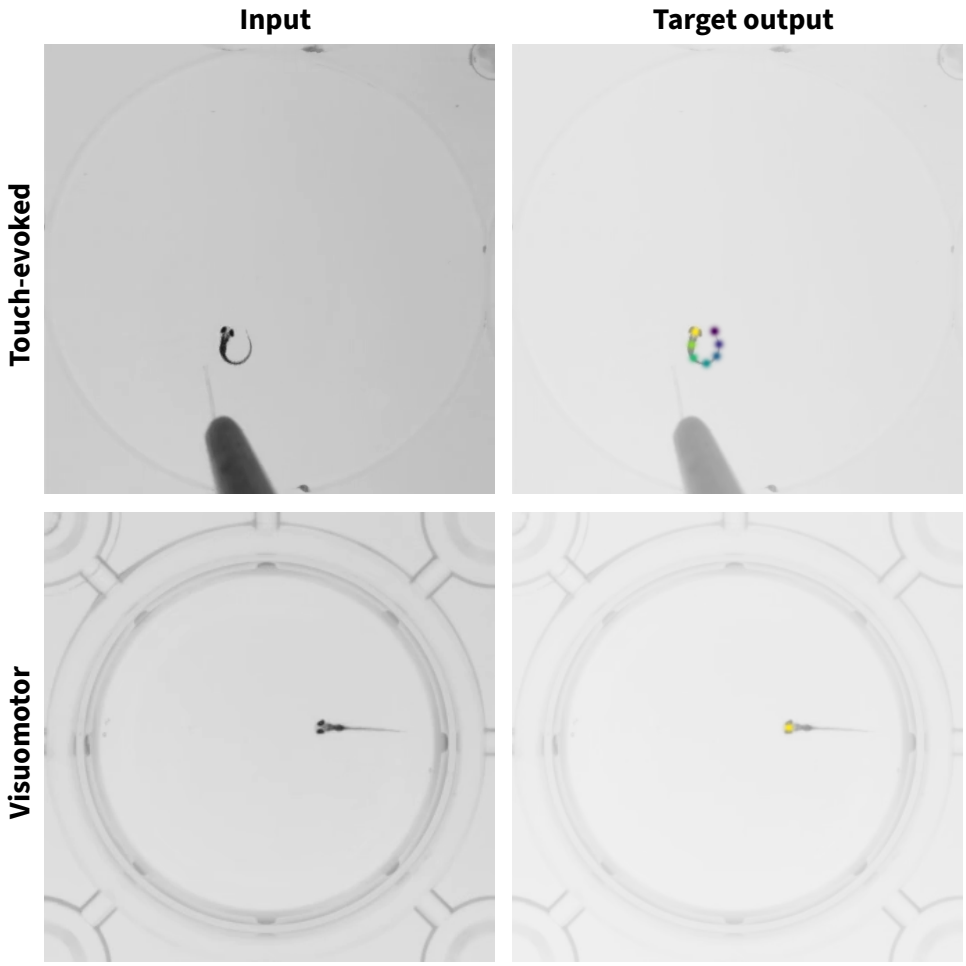

Figure 2. Examples of input–output pairs from the two datasets. Representative images from the touch-evoked response and visuomotor response datasets (left) paired with pseudocolored gaussians corresponding to labeled keypoints superimposed on the input images (right).

on mobile devices (A. Howard et al., 2019).

Despite its simplicity, the isometric architecture exhibited compelling advantages over its hourglass counterpart (Figure 4; Table 1). When trained on the touch-evoked response dataset, the isometric architecture crossed a mean squared error (MSE) loss threshold of 0.4 (roughly corresponding to a subjective minimum acceptable accuracy level) approximately 6× faster on average (Figure 4A). A similar result was observed for the visuomotor response dataset, although the relative magnitude of the difference was only approximately 3× (Figure 4B). Looking at other metrics, we observed that our isometric architecture achieved approximately 3× faster inference speed for the touch-evoked response dataset (Figure 4A) and approximately 2× faster inference speed for the visuomotor response dataset (Figure 4B). Although our isometric architecture contains a greater number of trainable parameters, its theoretical minimum memory footprint is considerably smaller (Table 1). Altogether, moving from an hourglass architecture to an isometric architecture

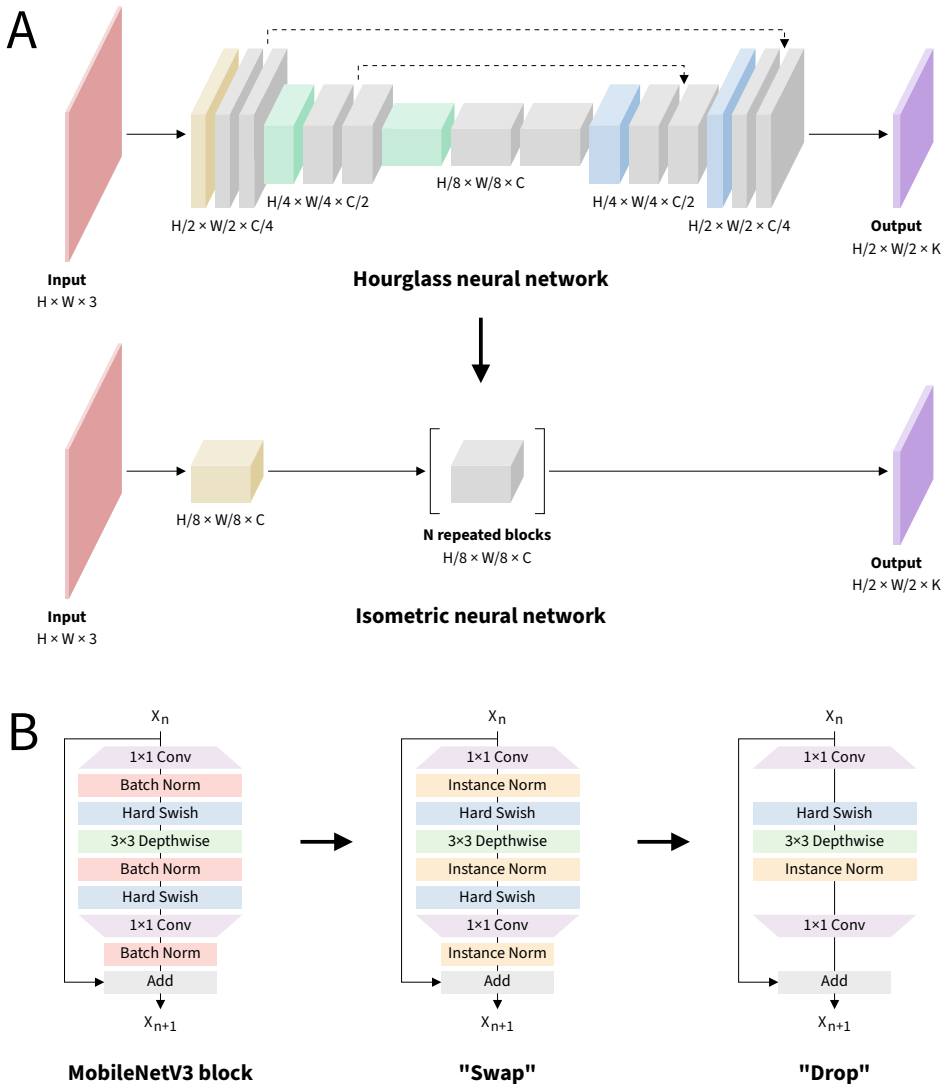

Figure 3. Overview of macro-architectural and micro-architectural design decisions leading to a simple, efficient, and easily customizable neural network architecture. (**A**) Schematic of our implementation of hourglass and isometric neural networks. Red indicates input images, yellow indicates intro blocks, green indicates downsampling blocks, blue indicates upsampling blocks, purple indicates outro blocks, and gray indicates residual blocks. Small solid arrows indicate the main flow of information through the neural networks, while small dashed arrows indicate long-range skip connections. The large solid arrow indicates the direction of our design preference. (**B**) Schematic of our implementation of the MobileNetV3 inverted bottleneck residual block and subsequent modifications. The large solid arrows indicate the direction of our design preferences.

appears to be a net positive design decision.

With our macro-architectural design strategy in place, we turned our attention to micro-architectural improvements (Figure 3B). The MobileNet family of neural networks, which introduced the MobileNetV3 block that acts as the repeated module in our isometric

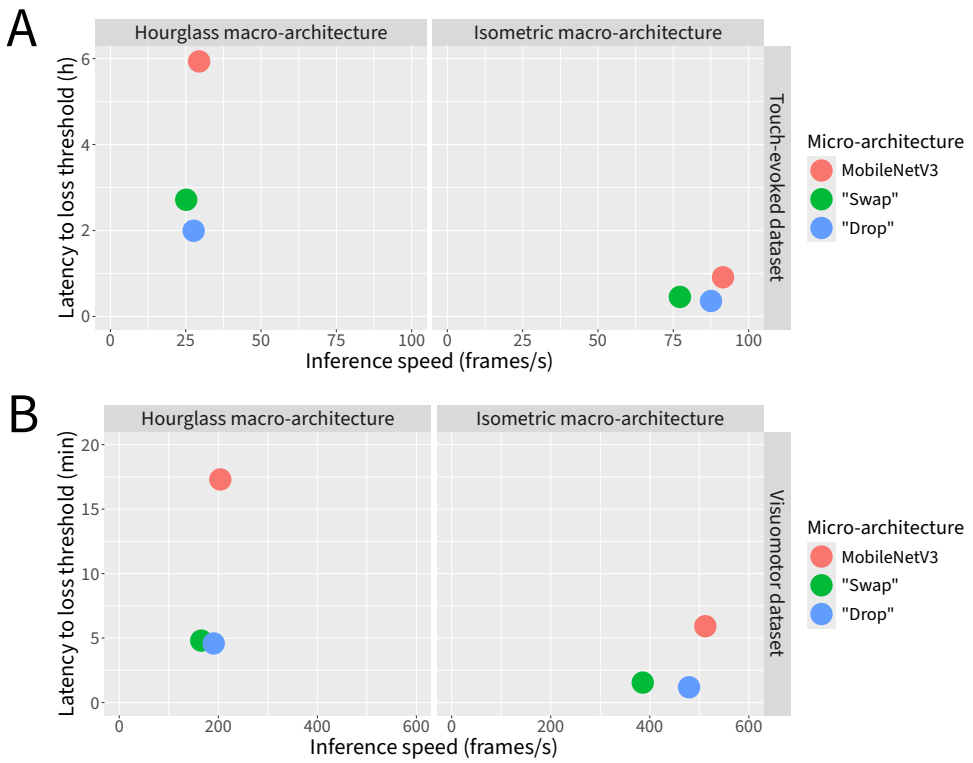

Figure 4. Through a series of macro-architectural and micro-architectural improvements, Marigold's neural networks achieve fast training and inference even on modest hardware. Neural networks were trained for each dataset, macro-architecture, and micro-architecture combination until crossing a mean squared error (MSE) loss threshold of 0.4. (**A**) Visualization of training speed vs. inference speed tradeoffs for the touch-evoked response dataset. (**A**) Visualization of training speed vs. inference speed tradeoffs for the visuomotor response dataset. Data represent the mean of 2–3 independent experiments in which the dataset was partitioned and the neural network weights were initialized using different random seeds.

architecture, were originally designed for fast inference on mobile devices (A. Howard et al., 2019; A. G. Howard et al., 2017; Sandler, Howard, et al., 2019). However, they were canonically trained on large-scale computer vision datasets on GPU or TPU clusters. As a result of this origin, we reasoned that the MobileNetV3 block may be better optimized for inference speed than training speed, particularly when training on small datasets.

This line of reasoning led us to explore alternatives to the Batch Normalization layers in the MobileNetV3 block (Figure 3B). Although the incorporation of Batch Normalization layers into neural networks can introduce considerable computational overhead, this is generally outweighed by the resulting improvement in training dynamics, which has resulted in their widespread use in computer vision architectures (Ioffe & Szegedy, 2015). At inference time, Batch Normalization layers introduce zero computational overhead as long as they can be "folded" into preceding convolutional layers. However, the effectiveness of Batch Normalization is strongly dependent on the batch size, with larger batch

Table 1. Parameter counts and training memory requirements for each dataset and neural network architecture combination. Training memory footprints represent the theoretical minimum memory requirement when training with a batch size of 8 (touch-evoked response dataset) or 32 (visuomotor response dataset).

Dataset	Macro-architecture	Micro-architecture	Trainable parameters	Training memory footprint (MB)
Touch-evoked	Hourglass	MobileNetV3	33,239	2,188.9
Touch-evoked	Hourglass	"Swap"	33,239	274.0
Touch-evoked	Hourglass	"Drop"	32,599	200.9
Touch-evoked	Isometric	MobileNetV3	118,064	801.0
Touch-evoked	Isometric	"Swap"	118,064	101.4
Touch-evoked	Isometric	"Drop"	116,784	73.3
Visuomotor	Hourglass	MobileNetV3	10,025	2,858.5
Visuomotor	Hourglass	"Swap"	10,025	89.4
Visuomotor	Hourglass	"Drop"	9,705	65.6
Visuomotor	Isometric	MobileNetV3	29,296	1,044.8
Visuomotor	Isometric	"Swap"	29,296	33.0
Visuomotor	Isometric	"Drop"	28,656	23.8

sizes generally yielding better results. This is problematic because the requirement to perform computations across the batch dimension results in a training memory footprint that increases linearly with the batch size, with the potential to exceed the limited amount of memory that web apps are allowed to allocate. By contrast, proposed alternatives to Batch Normalization, such as Layer Normalization (Ba et al., 2016), Instance Normalization (Ulyanov et al., 2017), and Group Normalization (Wu & He, 2018), do not require computations across the batch dimension. This enables the use of gradient accumulation to maintain a constant training memory footprint regardless of batch size. However, these alternatives cannot be "folded" into preceding convolutional layers, introducing computational overhead at inference time. In our experiments, Layer Normalization, Group Normalization, and Instance Normalization all led to faster training compared to Batch Normalization at a modest cost to inference speed (data not shown). However, "swapping" from Batch Normalization to Instance Normalization resulted in the most dramatic improvement and thus was incorporated into our final architectural design (Figure 3B; Figure 4).

Next, we asked whether it was possible to recover some of the inference speed that was lost by the switch from Batch Normalization to Instance Normalization without negatively impacting the training dynamics (Figure 3B). A logical way to accomplish this would be to "drop" some of the components in the MobileNetV3 block. We thus performed a series of ablation studies to determine which components were most dispensable. We were surprised to find that "dropping" two of the three Instance Normalization layers and one of the two Hard Swish layers actually led to more favorable training dynamics than any

other design iteration, while also fully recovering the inference speed (Figure **??**). We We note that the choices of which Instance Normalization layer and Hard Swish layer to keep were crucial in this design step, as some other combinations did not perform nearly as well (data not shown).

Altogether, our series of macro-architectural and micro-architectural design resulted in markedly faster training and inference speeds, with substantially reduced theoretical memory footprints, compared to our already reasonably efficient baseline. Although our final neural networks have considerably more trainable parameters, numbering in the tens of thousands to hundreds of thousands, compared to their baseline counterparts (Table 1), these numbers are still relatively small compared to many other lightweight neural networks which routinely contain several million parameters (e.g., A. Howard et al., 2019). Given that similar results were obtained for both datasets, our architecture appears to be flexible enough to handle a range of experimental paradigms. Additionally, whereas other architectures generally have interdependencies between aspects such as the input resolution, internal resolution, output resolution, number of residual blocks, and number of channels passed between blocks, the isometric nature of our architecture allows these aspects to be largely decoupled. It is thus relatively straightforward to customize our architecture for different experimental paradigms.

Training data requirements. Because manually labeling images is the most labor-intensive part of Marigold's workflow, we next asked whether it would be possible to obtain similar results with even less training data available. We investigated this by incrementally ablating images from the training split of each of our two datasets while controlling for the total number of parameter updates (Figure 5). To avoid compromising the ability to fairly evaluate each model's ability to generalize to previously unseen data, we held the number of images in the validation split constant.

Across both datasets, ablating half of the training images (leaving 192 and 96 images for the touch-evoked response and visuomotor response datasets, respectively) had a minimal effect on the final validation loss. Ablating three quarters of the training images (leaving 96 and 48 images for the visuomotor response datasets, respectively) resulted in a noticeably more detrimental effect on the final validation loss, particularly for the touch-evoked response dataset. While we conclude that our neural networks can be trained effectively even with limited training data (particularly given that our datasets contain several orders of magnitude fewer images than many conventional computer vision datasets), it is likely that our data augmentation pipeline could be improved to yield better performance when datasets are even smaller.

WebAssembly implementation. Having optimized our neural network architecture using our PyTorch-based implementation and verified that it can be effectively trained with relatively little training data, we next focused on implementing this functionality within

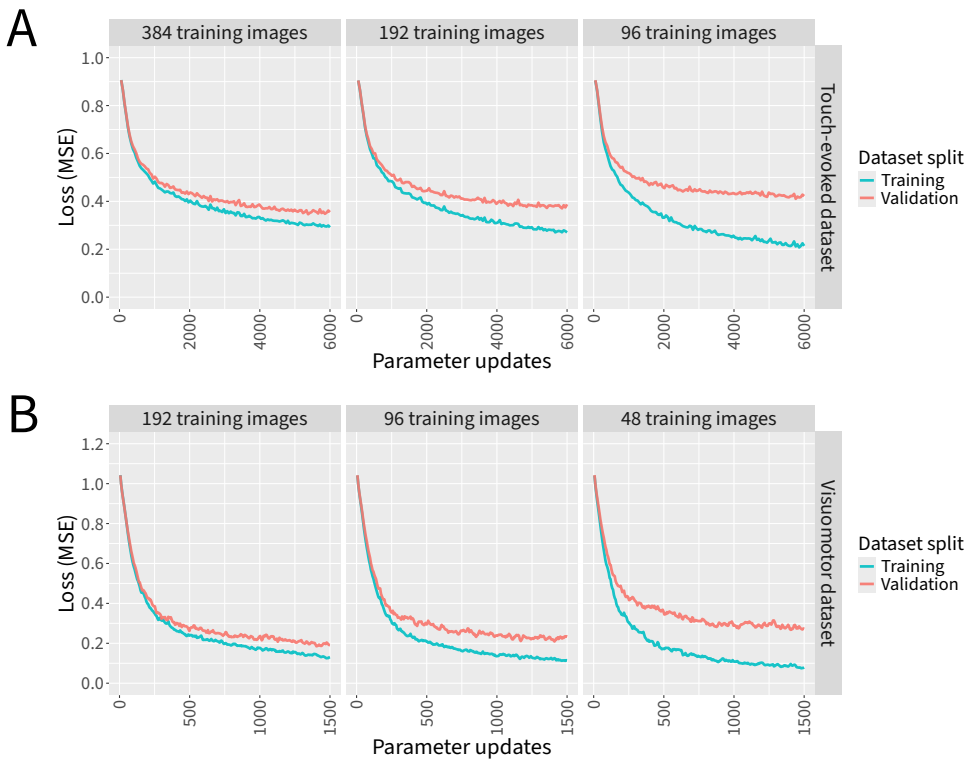

Figure 5. Marigold's neural networks can be trained effectively even with limited training data. Neural networks were trained on increasingly ablated sets of training images while maintaining a constant number of parameter updates and evaluating performance on a constant number of validation images. (**A**) Training and validation loss curves for the touch-evoked response dataset. (**B**) Training and validation loss curves for the visuomotor response dataset. Data represent the mean of 3–4 independent experiments in which the dataset was partitioned and the neural network weights were initialized using different random seeds.

our web app. To achieve this, we implemented the necessary neural network layer and training operations in C++ and used this to generate WebAssembly, which allows such code to run within a browser at speeds that are competitive with native C++ code (Haas et al., 2017; Perkel, 2024). To evaluate the effectiveness of this approach, we compared the resulting training and inference speeds to those of our PyTorch implementation (Table 2). In terms of inference speed, our WebAssembly-based implementation achieves slightly better or slightly worse than half the speed of PyTorch for the touch-evoked response and visuomotor response datasets, respectively. A similar result was obtained in terms of training speed for the touch-evoked response dataset. Interestingly, our WebAssembly-based implementation closely matched the speed of PyTorch for training our visuomotor response dataset. Overall, our implementation achieves speeds that are competitive on CPU with the highly optimized PyTorch library. This is particularly noteworthy given that web apps are permitted only limited access to memory and CPU features.

Table 2. Training and inference speeds for PyTorch and Marigold implementations of neural networks for each dataset. Measurements reflect single-threaded CPU performance. Training speeds were measured with a batch size of 8 (touch-evoked response dataset) or 32 (visuomotor response dataset). Inference speeds were measured with a batch size of 1. Data reflects the mean of 5 independent measurements.

Dataset	Implementation	Training speed (images/s)	Inference speed (images/s)
Touch-evoked	PyTorch	18.0	87.5
Touch-evoked	Marigold	11.6	34.8
Visuomotor	PyTorch	108.1	479.3
Visuomotor	Marigold	108.7	324.7

Effects of genotype and touch location in *slc1a2b*$^{tk57/tk57}$ and wild-type sibling touch-evoked responses. Using Marigold, we investigated novel aspects of the *techno trousers* (*tnt*) locomotor phenotype (Granato et al., 1996; McKeown et al., 2012). This mutant, henceforth referred to as *slc1a2b*$^{tk57/tk57}$, harbors a loss-of-function missense mutation in *slc1a2b* (Ensembl ID: ENSDARG00000102453) located on chromosome 25, which encodes Eaat2b, a glutamate transporter predominantly expressed in astroglia (Hotz et al., 2022; McKeown et al., 2012). We note that this gene and its protein product have also been referred to as *slc1a2a* and Eaat2a, respectively (Gesemann et al., 2010; Hotz et al., 2022); however, current genome databases indicate that these particular identifiers refer to a distinct gene found on chromosome 7 (Ensembl ID: ENSDARG00000052138), and its protein product (Breuer et al., 2019). *slc1a2b*$^{tk57/tk57}$ embryos demonstrate a hyperactive response to touch by 48 hpf; however, the larvae are effectively paralyzed and shorter along the rostral–caudal axis by 96 hpf (McKeown et al., 2012). Small twitch responses to touch can be observed at 96 hpf, suggesting the lack of response is due to motor issues rather than sensory issues. Because the auditory and visual systems are not fully developed by 96 hpf, tactile stimulation at earlier time points is the most robust way to elicit this hyperactive phenotype.

Reticulospinal neurons, such as the Mauthner cell and its homologs (collectively referred to as the Mauthner array), are the primary mediators of the short latency touch-evoked response. Head-stimulated responses recruit the full Mauthner array while tail-stimulated responses recruit the Mauthner cell alone, resulting in kinematically distinct responses based upon touch location (K. S. Liu & Fetcho, 1999). By comparing touch-evoked responses of head- and tail-stimulated *slc1a2b*$^{tk57/tk57}$ embryos, we reasoned that we could uncover novel aspects of the *slc1a2b*$^{tk57/tk57}$ locomotor phenotype. Here, we seek to use Marigold's advanced capabilities in analyzing zebrafish locomotion to (1) append new information to what is known about *slc1a2b*$^{tk57/tk57}$ hyperactivity, and (2) uncover possible differences between the kinematics of head- and tail-stimulated *slc1a2b*$^{tk57/tk57}$ startle responses.

Following head stimulation, wild-type embryos reliably exhibit high amplitude bends

followed by lower amplitude undulations, reorienting the embryo and allowing it to swim away from the perceived threat (Figure 6A). Head-stimulated *slc1a2b*$^{tk57/tk57}$ embryos swim significantly longer and farther than wild-type controls (two-way ANOVA, $F(1,86) = 59.58$, $p < 0.001$ and $F(1,86) = 57.81$, $p < 0.001$ for duration and distance, respectively) (Figure 6B–C). Consistent with previous literature (McKeown et al., 2012), *slc1a2b*$^{tk57/tk57}$ embryos perform more high amplitude body bends during the escape response (two-way ANOVA, $F(1,86) = 35.88$, $p < 0.001$) (Figure 6D). However, the number of high amplitude bends performed between head- and tail-stimulated embryos did not vary by genotype (two-way ANOVA, $F(1,86) = 0.00$, $p >= 0.05$). The distribution of high amplitude body bends throughout the startle response differs markedly between *slc1a2b*$^{tk57/tk57}$ and wild-type embryos. *slc1a2b*$^{tk57/tk57}$ embryos perform extended stretches of high amplitude bends throughout the responses (Figure 6E; Figure 7A). Of all embryos that perform a high amplitude bend, the distribution of *slc1a2b*$^{tk57/tk57}$ high amplitude bends features more bends occurring later in the response (Figure 7A–B).

We further investigated characteristics of the first body bend within each startle response. Regardless of genotype, the initial bend of tail-stimulated embryos is significantly lower in amplitude than that of head-stimulated embryos, often not crossing the 110° threshold (two-way ANOVA, $F(1,86) = 79.703$, $p < 0.001$) (Figure 7C). Tail-stimulated embryos also reached their maximum angle significantly earlier in the response (two-way ANOVA, $F(1,86) = 44.833$, $p < 0.001$) (Figure 7D). Interestingly, we note genotype- and touch location-dependent effects on the maximum body angle of the startle response. Among wild-type embryos, the maximum body bend had a significantly lower amplitude in tail-stimulated embryos than in head-stimulated embryos (two-way ANOVA, $F(1,86) = 36.78$, $p < 0.001$). In contrast, the maximum angle for tail-stimulated *slc1a2b*$^{tk57/tk57}$ responses is nearly indistinguishable from head-stimulated *slc1a2b*$^{tk57/tk57}$ responses (two-way ANOVA interaction, $F(1,86) = 14.45$, $p < 0.001$) (Figure 7E). To determine whether embryos execute the maximum angle beyond the initial bend, we measured the time required for the embryo to reach the maximum body angle of the response. *slc1a2b*$^{tk57/tk57}$ embryos perform their maximum body bend significantly later in the response than wild-type embryos (two-way ANOVA, $F(1,86) = 25.26$, $p < 0.001$) (Figure 7F).

Effects of developmental stage and feeding on larval visuomotor response behavior. To demonstrate additional capabilities of Marigold, particularly its ability to accurately and efficiently analyze the behavior of larval zebrafish in multiwell plates, we examined the effects of developmental stage and feeding on the larval zebrafish visuomotor response (Brockerhoff et al., 1997; Burgess & Granato, 2007; Lee et al., 2019; Randlett et al., 2019). Larvae were fed or not fed once daily from 5 dpf through 7 dpf and behavior was recorded in 24-well plates 4–6 h after feeding at 5 dpf and 7 dpf. To examine the effects of age and feeding across multiple behavioral modes, we recorded fish during periods of dark adaptation, light stimulus, light adaptation, and dark stimulus (Figure 8A).

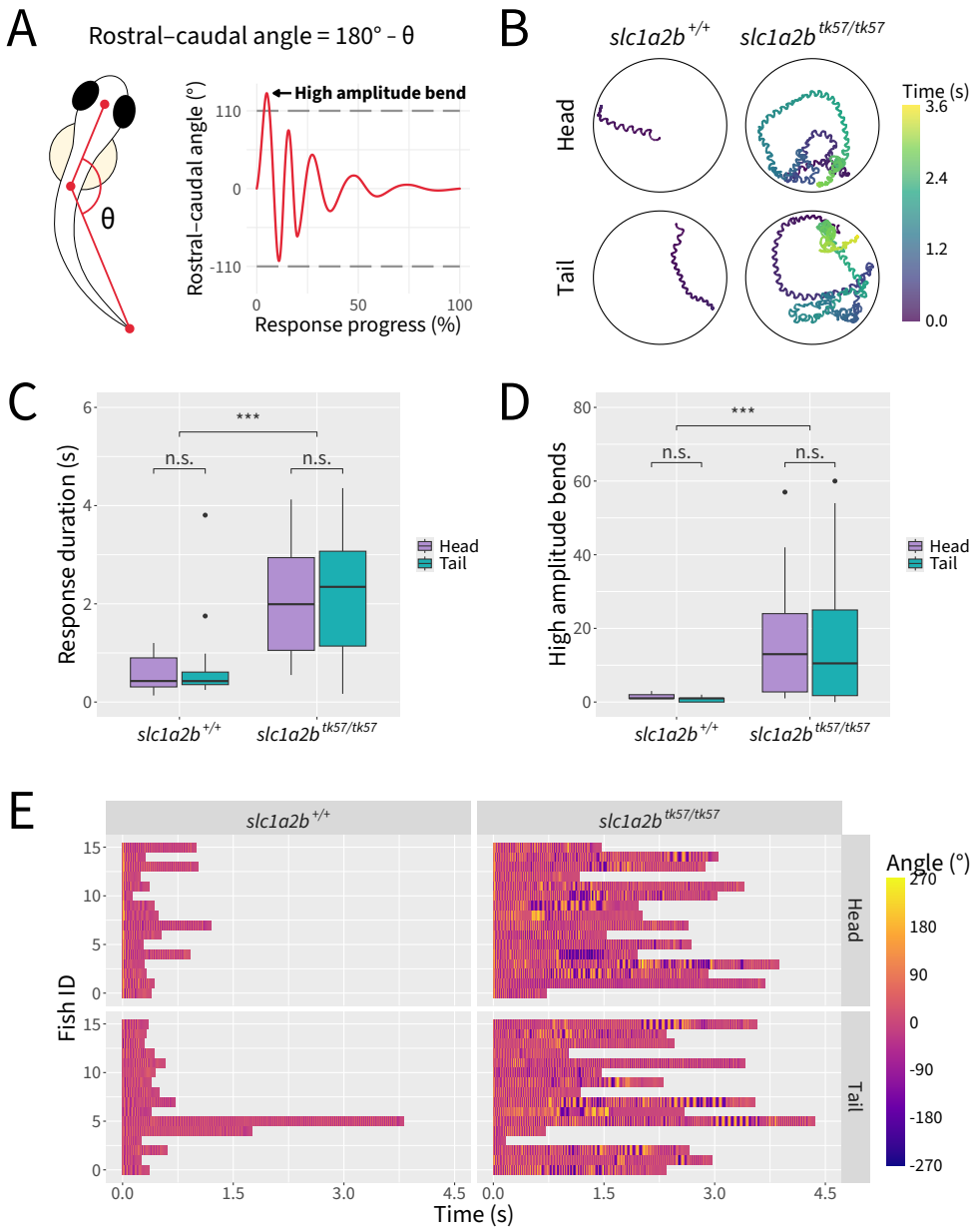

Figure 6. *slc1a2b*$^{tk57/tk57}$ mutant zebrafish exhibited hyperactive touch-evoked startle behavior by multiple kinematic measures, indiscriminate of the touch location. (**A**) Schematic illustrating calculation of rostral–caudal angle and quantification of high amplitude body bends. (**B**) Representative trajectories for each touch location and genotype, color-mapped by timestamp. (**C-D**) Quantification of (C) response duration and (D) high amplitude bends for each touch location and genotype. (**E**) Visualization of individual rostral–caudal angles over time for 16 representative fish from each touch location and genotype. Each row represents the startle response of a single fish. Colormap corresponds to the rostral–caudal angle at a given point in time. Statistical significance determined by two-way ANOVA with Tukey ; n.s.: not significant; ***: $p < 0.001$; n = 21-24 fish per condition across 3 independent experiments.

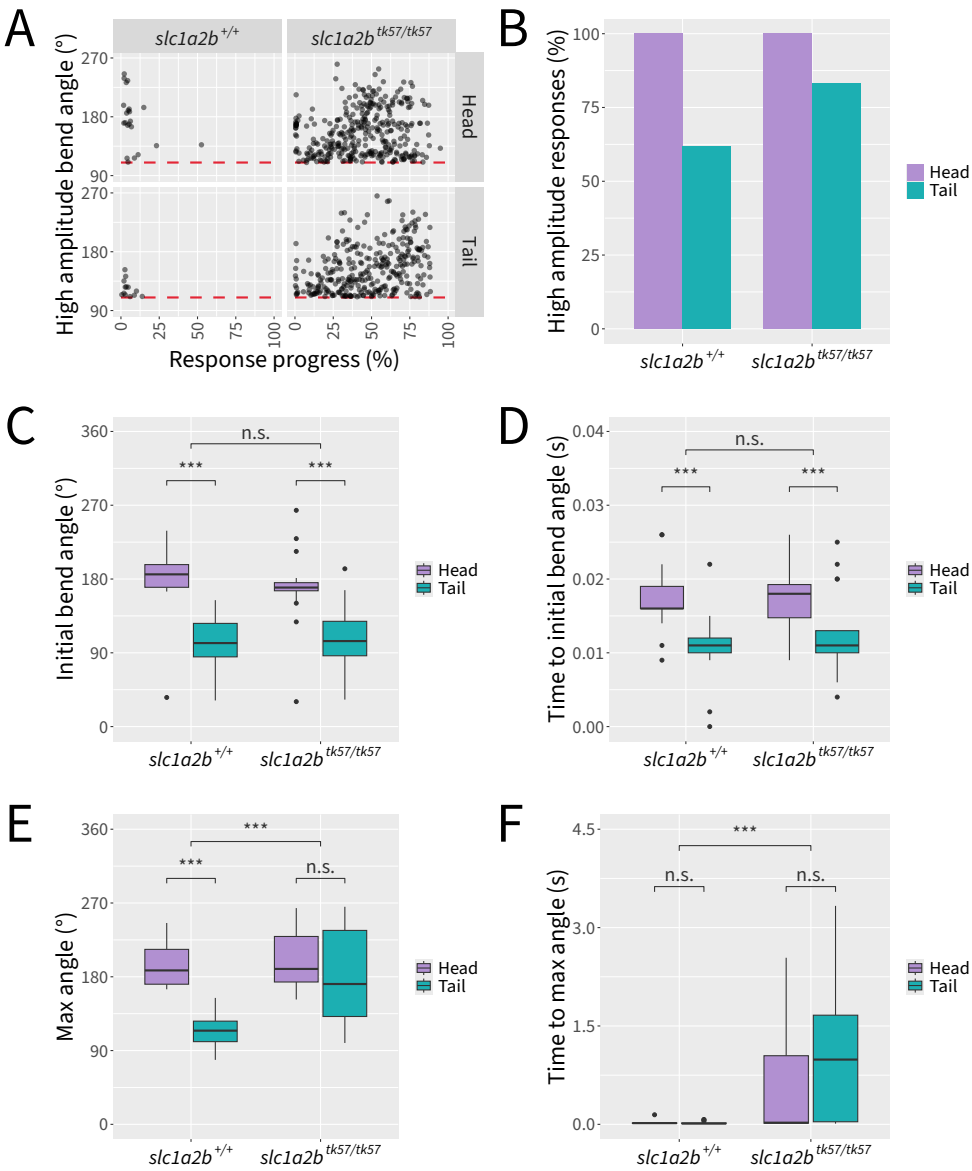

Figure 7. *slc1a2b*$^{tk57/tk57}$ and wild-type embryos performed rostral–caudal body bends dependent on interactions between genotype and touch location. (**A**) Visualization of absolute rostral–caudal angle magnitudes during high amplitude bends, depicted over time according to corresponding response progress. Data represents 16 representative fish per genotype and touch location. Red line indicates the threshold of 110° for classification as a high amplitude bend. (**B**) Percentage of responses which contain one or more high amplitude bends. (**C–F**) Quantification of (C) peak angle of first bend, (D) time to peak angle of first bend, (E) maximum rostral–caudal angle, and (F) time to maximum rostral–caudal angle. Statistical significance determined by two-way ANOVA with Tukey post hoc; n.s.: not significant; ***: $p < 0.001$; n = 21-24 fish per condition across 3 independent experiments.

Visualization of fish trajectories during the four recording periods suggested reduced swimming in unfed fish, particularly at 7 dpf (Figure 8B). Smoothed mean speed traces

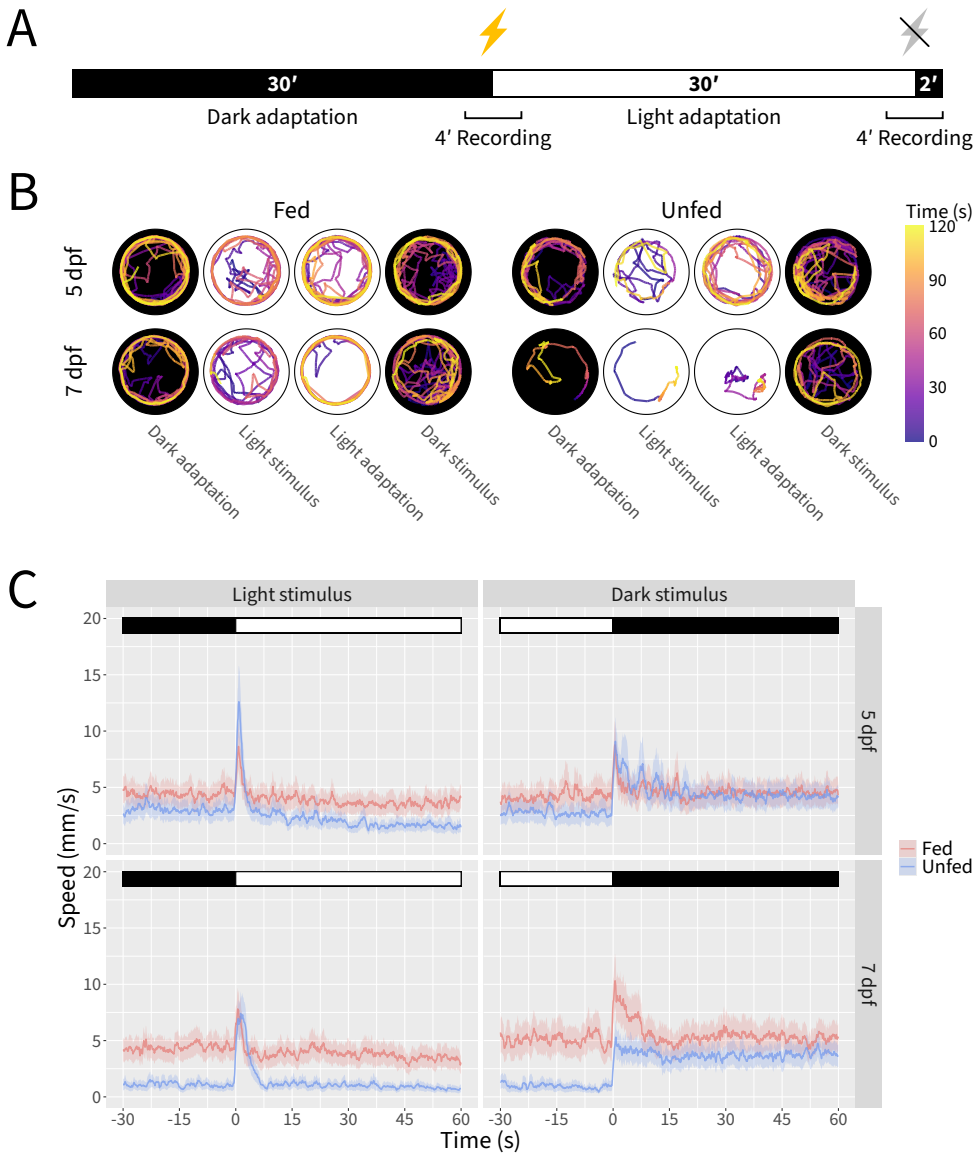

Figure 8. Visuomotor response trajectory traces and speed plots display reduced swimming in unfed larvae. (**A**) Schematic of visuomotor response recording paradigm. (**B**) Representative trajectory traces for fed and unfed fish at 5 and 7 dpf, during periods of dark adaptation, light stimulus, light adaptation, and dark stimulus. (**C**) Speed plots for fed and unfed fish at 5 and 7 dpf, during periods of dark adaptation, light stimulus, light adaptation, and dark stimulus. Traces represent the mean (lines) plus and minus the standard error of the mean (ribbons) after applying a mean filter with a sliding window corresponding to 1 s (20 frames). Data represents n = 44–46 fish per condition across 3 independent experiments.

provided further insight into this pattern (Figure 8C). During dark adaptation, both fed and unfed fish maintained stable activity levels, however this baseline activity level was lower in unfed fish, with the effect becoming more pronounced at 7 dpf. Interestingly,

the baseline activity level was essentially identical across fed fish regardless of age. Immediately following the light stimulus, both fed and unfed fish mobilized a rapid burst of activity before returning to the baseline activity level. Fed and unfed fish exhibited similar level of activity during light adaptation and dark adaptation, with unfed fish again showing a lower level of activity which was more pronounced at 7 dpf. Strikingly, whereas the acute reaction to the dark stimulus was followed by a return to baseline in fed fish (albeit more gradually than was observed following the light stimulus), unfed fish maintained a heightened level of activity which resembled the baseline activity level of fed fish.

Analysis of maximum speed levels during the 5 seconds immediately following light or dark stimuli revealed no significant differences or interactions between age and feeding status for light stimuli (two-way ANOVA, $F(1,174) = 3.537$ for age, 1.608 for diet, and 2.401 for interaction, $p >= 0.05$ in all three cases). In contrast, immediately following dark stimuli unfed fish exhibited an attenuated response at 7 dpf, but not at 5 dpf (Figure 8C; Figure 9A), reflecting a main effect of diet (two-way ANOVA, $F(1,174) = 21.659$, $p < 0.001$) and an interaction between age and diet (two-way ANOVA, $F(1,174) = 29.881$, $p < 0.001$), but no main effect of age (two-way ANOVA, $F(1,174) = 0.107$, $p >= 0.05$). Further insights were gleaned by examining the number of swim bouts, which we define as beginning when a fish's speed exceeds 2 mm/s and ending when the fish's speed drops below this threshold. Here, a main effect of diet was consistently observed during periods of dark adaptation, light stimulus, light adaptation, and dark stimulus (two-way ANOVA, $F(1,178) = 144.94$, 100.513, 48.606, and 27.127, respectively, $p < 0.001$ during all four periods). However, a significant main effect of age was observed during only the first three of these periods (Figure 9B) (two-way ANOVA, $F(1,178) = 23.61$, 11.855, 9.129, and 3.535, respectively, $p < 0.001$, $p < 0.001$, $p < 0.01$, and $p >= 0.05$, respectively). The interaction between age and diet followed a similar pattern (Figure 9B) (two-way ANOVA, $F(1,178) = 26.08$, 9.942, 8.146, and 0.504, respectively, $p < 0.001$, $p < 0.01$, $p < 0.01$, and $p >= 0.05$, respectively). In addition to the number of swim bouts, we examined the distributions of swim bout durations (Figure 9C). These distributions largely overlapped for fed and unfed fish at 5 dpf during all four observation periods. However, while the distributions for fed fish were stable across both developmental stages, the distributions for unfed fish exhibited a shift toward swim bouts with shorter durations at 7 dpf. This shift was observed during periods of dark adaptation, light stimulus, and light adaptation, but not dark stimulus.

Discussion

Example applications. To demonstrate the utility of Marigold, we analyzed touch-evoked responses in *slc1a2b*$^{tk57/tk57}$ embryos and their wild type siblings, with the stimulus being applied to either the head or tail. We first recapitulated known phenotypic differences in the touch-evoked response between *slc1a2b*$^{tk57/tk57}$ embryos and their wild type siblings, including increases in response duration and the number of high amplitude body bends in *slc1a2b*$^{tk57/tk57}$ embryos. Additionally, we explored whether applying a stimulus to either

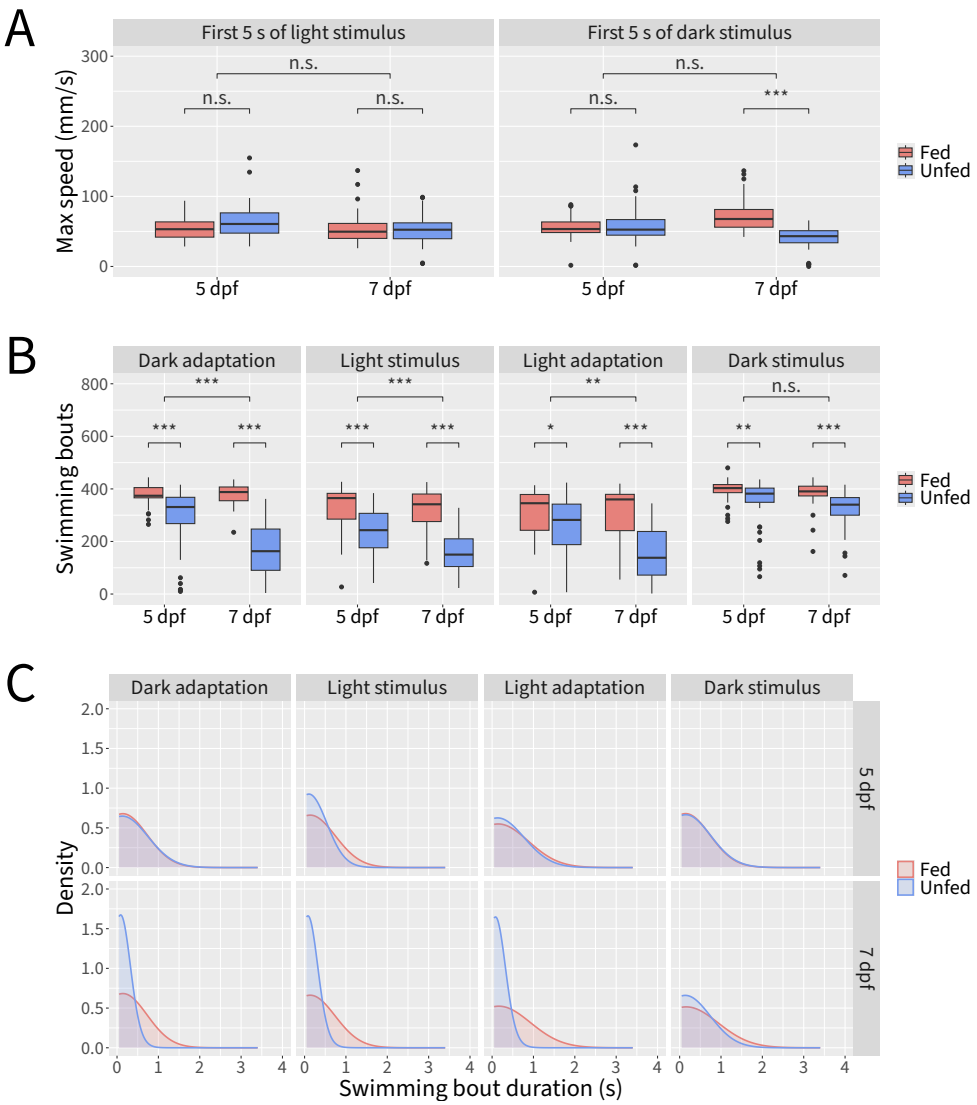

Figure 9. Detailed behavioral profiling reveals effects and interactions of age and diet on larval visuomotor responses. (**A**) Quantification of maximum speed during the first 5 s of the response to light and dark stimuli for fed and unfed fish at 5 and 7 dpf. (**B**) Quantification of swimming bout number for fed and unfed fish at 5 and 7 dpf, during periods of dark adaptation, light stimulus, light adaptation, and dark stimulus. (**C**) Visualization of swimming bout duration distributions as kernel density estimates for fed and unfed fish at 5 and 7 dpf, during periods of dark adaptation, light stimulus, light adaptation, and dark stimulus. Statistical significance determined by two-way ANOVA with Tukey post hoc; n.s.: not significant; *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$; n = 44–46 fish per condition across 3 independent experiments.

the head or tail would further reveal distinct kinematic profiles in *slc1a2b*$^{tk57/tk57}$ embryos. Within tail-stimulated responses, fewer embryos demonstrate an initial bend that passes the high amplitude threshold, compared to head-stimulated embryos, regardless of genotype. Because the embryo is already oriented away from the perceived threat when

touched on the tail, reorientation via a high amplitude bend may be less advantageous in this scenario.

We further show that while there are stimulus location differences in the maximum rostral–caudal angle that a wild-type embryo performs, those differences are lost in *slc1a2b*$^{tk57/tk57}$ embryos. Both head- and tail-stimulated *slc1a2b*$^{tk57/tk57}$ embryos reach the maximum rostral–caudal angle later in the response than wild-type embryos, which could explain why no touch location differences were found for the maximum angle of *slc1a2b*$^{tk57/tk57}$ responses. Although *slc1a2b* is not expressed in the Mauthner cell, *slc1a2b*-expressing glial cells surround the Mauthner array at 48 hpf (Hotz et al., 2022; McKeown et al., 2012). Based on the data presented here, we propose that the disrupted kinematic profile of *slc1a2b*$^{tk57/tk57}$ embryos may be due to a reactivation of escape circuitry late into the response.

Interestingly, *slc1a2b* mutants harboring nonsense mutations are able to swim spontaneously at 5 dpf (Hotz et al., 2022), whereas the *slc1a2b*$^{tk57/tk57}$ allele described here and in McKeown et al., 2012 is fully paralyzed by 96 hpf. This difference could be explained by the nature of the mutations; premature termination codons generated using CRISPR-Cas9 have been shown to trigger genetic compensation via transcriptional adaptation (El-Brolosy et al., 2019; Sztal & Stainier, 2020). Additionally, the *slc1a2b*$^{tk57/tk57}$ allele described here is maintained on a TL background, while Hotz et al., 2022 report using WIK and Tübingen wild-type strains. As notable differences have been reported between wild-type zebrafish strains (Audira et al., 2020; Maximino et al., 2013), this could also explain the difference in phenotypic severity. Lastly, Hotz et al., 2022 describe a mutation in exon 3 of *slc1a2b* while the mutation underlying *slc1a2b*$^{tk57/tk57}$ is located in exon 7 of *slc1a2b* (McKeown et al., 2012).

We also used Marigold to examine visuomotor responses in wild-type larvae at two developmental stages and in fed and unfed states. Larval responses to the visuomotor response differ across developmental time (de Esch et al., 2012; Y. Liu et al., 2015). The first zebrafish larvae startle responses to visual stimulation can be observed around 3 dpf (Easter & Nicola, 1996). By 5 dpf, larvae inflate their swim bladders, exhibit robust swimming behavior, and actively hunt for food. Zebrafish larvae utilize nutrients from their yolks until at least 7 dpf, and it has been shown that feeding can be delayed until 8 dpf without negatively affecting juvenile survival or growth (Hernandez et al., 2018; Lucore & Connaughton, 2021; Schwartz et al., 2021). A previous study examining the effects of feeding on larval swim behavior, visual stimuli avoidance, and inter-fish distance reported significant behavioral differences between fed and unfed fish at 6 and 7 dpf, but no significant effects at 5 dpf (Clift et al., 2014). This observation led to the recommendation that 5 dpf generally be preferred for behavioral experiments. However, the effect of feeding has not previously been evaluated with respect to visuomotor responses.

While our data confirm that feeding status has a more pronounced effect on visuomotor response behavior at 7 dpf than at 5 dpf, we also provide evidence for significant effects

of feeding on behavioral parameters even at 5 dpf (Brockerhoff et al., 1997; Burgess & Granato, 2007; Lee et al., 2019; Randlett et al., 2019). Thus, our results highlight the importance of reporting and controlling for developmental stage and feeding status in visuomotor response assays.

In both sets of experiments, the in-depth behavioral profiling facilitated by Marigold led to novel findings, despite the *slc1a2b*$^{tk57/tk57}$ mutant having been previously characterized and visuomotor responses being a widely adopted experimental paradigm.

Advantages, limitations, and future directions. To our knowledge, Marigold is the first free and open source software to perform machine learning-based animal pose tracking for behavioral analysis entirely in-browser, an approach which allows us to make this type of analysis considerably more accessible to both students and researchers. By implementing our software as a web app, we are able to take advantage of the web platform to deliver a rich user interface supporting a highly streamlined workflow, all without requiring any installation. The web app implementation also allows Marigold to run entirely within the security sandbox provided by the user's web browser. Moreover, by designing a neural network architecture from the ground up to perform highly efficient animal pose estimation and implementing this architecture using the recently introduced WebAssembly technology (Haas et al., 2017; Perkel, 2024), we are able to achieve competitive training and inference speeds even on modestly powered computers lacking a dedicated GPU and despite the limited access to memory and computational resources web apps are afforded.

Marigold's ability to allow the user to create their own dataset and train custom models, while maintaining a focus on functionality that is of particular interest to the zebrafish research community, makes the program applicable to a broad range of zebrafish behavioral experiments. For instance, in addition to the applications demonstrated here (i.e., analysis of touch-evoked and visuomotor responses), we also have successfully used Marigold to analyze spinalized larvae, head-embedded larvae, and adult fish (data not shown).

Marigold is currently limited to analyzing a single animal in each region of interest (ROI), a decision made to maintain the program's simplicity and our focus on embryonic and early larval stages of zebrafish development, when social interaction is minimal (Stednitz & Washbourne, 2020). However, future work could extend Marigold to support the analysis of multiple fish in each ROI and for three-dimensional pose tracking, since both social interaction and three-dimensional swimming become more prominent during late larval and juvenile development (Macrì et al., 2017; Stednitz & Washbourne, 2020). Additionally, while we have successfully used Marigold for pose tracking of other animal species such as mice (data not shown), there is clearly an opportunity to more formally extend the web app-based approach of Marigold for pose tracking in other species and to adapt it for other types of behavioral analysis.

Conclusions

Much of the existing software used for zebrafish behavioral analysis have constraints that limit their utility, including being cost prohibitive or requiring substantial programming experience. Here, we describe Marigold, which aims to address these limitations. It is a free and open source web app that utilizes machine learning to perform zebrafish pose tracking. Marigold features efficient neural networks that allow for competitive training and inference speeds even on basic laptop computers. We demonstrate the utility of Marigold by using it to uncover novel aspects of the *tnt* mutant touch-evoked escape response and of the effects of feeding and developmental stage on the larval visuomotor response. We expect that Marigold will serve as a user-friendly tool to aid the zebrafish community in conducting robust, high-throughput behavioral analysis.

Acknowledgments

We thank Hanna E. Dorman Barclay and Caroline Martin for excellent fish care. We thank all members of the Downes Lab, as well as the University of Massachusetts Amherst and Amherst College zebrafish communities, for thoughtful discussion. We are extremely grateful to Scott Alfeld for helpful discussion and feedback on an early version of the manuscript.

Author contributions

GT, JGT, and GBD acquired funding for the project. GT, RMR, WB, JGT, and GBD conceptualized the software and the biological experiments. GT conceptualized and conducted the computational experiments and developed the software. GT and GM collected the training data and conducted the biological experiments. RMR and BEC labeled the training data. GT, RMR, and WB analyzed the biological experiments and prepared the figures. GT, RMR, WB, BEC, GBD, and JGT wrote the manuscript. All authors read and approved the final manuscript.

Data availability

The datasets used and/or analyzed during this study are available from the corresponding authors upon reasonable request. The source code for Marigold is licensed under the GNU General Public License version 3 (GPLv3) or later and is available at: https://github.com/downeslab/marigold. The Marigold web app itself is available at: https://downeslab.github.io/marigold/.

Funding

This work was funded by the National Science Foundation (IOS 1456866) to GBD and JGT, the American Epilepsy Society (AES2017SD) to GBD, and the University of Massachusetts Amherst Biology Department (2019 Distinguished Graduate Student Summer Research Fellowship) to GT.

Competing interests

All authors declare that they have no competing interests.

References

Audira, G., Siregar, P., Strungaru, S.-A., Huang, J.-C., & Hsiao, C.-D. (2020). Which Zebrafish Strains Are More Suitable to Perform Behavioral Studies? A Comprehensive Comparison by Phenomic Approach [Number: 8]. *Biology*, *9*(8), 200. https://doi.org/10.3390/biology9080200

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016, July). Layer Normalization [arXiv: 1607.06450 [cs, stat]]. https://doi.org/10.48550/arXiv.1607.06450

Baraban, S. C., Dinday, M. T., & Hortopan, G. A. (2013). Drug screening in Scn1a zebrafish mutant identifies clemizole as a potential Dravet syndrome treatment [Number: 1]. *Nature Communications*, *4*(1), 2410. https://doi.org/10.1038/ncomms3410

Berg, E. M., Björnfors, E. R., Pallucchi, I., Picton, L. D., & El Manira, A. (2018). Principles Governing Locomotion in Vertebrates: Lessons From Zebrafish. *Frontiers in Neural Circuits*, *12*, 73. https://doi.org/10.3389/fncir.2018.00073

Breuer, M., Guglielmi, L., Zielonka, M., Hemberger, V., Kölker, S., Okun, J. G., Hoffmann, G. F., Carl, M., Sauer, S. W., & Opladen, T. (2019). QDPR homologues in Danio rerio regulate melanin synthesis, early gliogenesis, and glutamine homeostasis [Number: 4]. *PLOS ONE*, *14*(4), e0215162. https://doi.org/10.1371/journal.pone.0215162

Brockerhoff, S. E., Hurley, J. B., Niemi, G. A., & Dowling, J. E. (1997). A new form of inherited red-blindness identified in zebrafish [Number: 11]. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, *17*(11), 4236–4242.

Burgess, H. A., & Granato, M. (2007). Modulation of locomotor activity in larval zebrafish during light adaptation [Number: Pt 14]. *The Journal of Experimental Biology*, *210*(Pt 14), 2526–2539. https://doi.org/10.1242/jeb.003939

Clift, D., Richendrfer, H., Thorn, R. J., Colwill, R. M., & Creton, R. (2014). High-Throughput Analysis of Behavior in Zebrafish Larvae: Effects of Feeding [Number: 5]. *Zebrafish*, *11*(5), 455–461. https://doi.org/10.1089/zeb.2014.0989

d'Amora, M., & Giordani, S. (2018). The Utility of Zebrafish as a Model for Screening Developmental Neurotoxicity. *Frontiers in Neuroscience*, *12*, 976. https://doi.org/10.3389/fnins.2018.00976

de Esch, C., van der Linde, H., Slieker, R., Willemsen, R., Wolterbeek, A., Woutersen, R., & De Groot, D. (2012). Locomotor activity assay in zebrafish larvae: Influence of age, strain and ethanol [Number: 4]. *Neurotoxicology and Teratology*, *34*(4), 425–433. https://doi.org/10.1016/j.ntt.2012.03.002

Easter, S. S., & Nicola, G. N. (1996). The development of vision in the zebrafish (Danio rerio) [Number: 2]. *Developmental Biology*, *180*(2), 646–663. https://doi.org/10.1006/dbio.1996.0335

El-Brolosy, M. A., Kontarakis, Z., Rossi, A., Kuenne, C., Günther, S., Fukuda, N., Kikhi, K., Boezio, G. L., Takacs, C., Lai, S.-L., Fukuda, R., Gerri, C., Giraldez, A. J., & Stainier, D. Y. (2019). Genetic compensation triggered by mutant mRNA degradation [Number: 7751]. *Nature*, *568*(7751), 193–197. https://doi.org/10.1038/s41586-019-1064-z

Gesemann, M., Lesslauer, A., Maurer, C. M., Schönthaler, H. B., & Neuhauss, S. C. (2010). Phylogenetic analysis of the vertebrate Excitatory/Neutral Amino Acid Transporter (SLC1/EAAT) family reveals lineage specific subfamilies [Number: 1]. *BMC Evolutionary Biology*, *10*(1), 117. https://doi.org/10.1186/1471-2148-10-117

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 249–256. https://proceedings.mlr.press/v9/glorot10a.html

Gore, S. V., Kakodkar, R., Del Rosario Hernández, T., Edmister, S. T., & Creton, R. (2023). Zebrafish Larvae Position Tracker (Z-LaP Tracker): A high-throughput deep-learning behavioral approach for the identification of calcineurin pathway-modulating drugs using zebrafish larvae [Number: 1]. *Scientific Reports*, *13*(1), 3174. https://doi.org/10.1038/s41598-023-30303-w

Granato, M., van Eeden, F., Schach, U., Trowe, T., Brand, M., Furutani-Seiki, M., Haffter, P., Hammerschmidt, M., Heisenberg, C., Jiang, Y., Kane, D., Kelsh, R., Mullins, M., Odenthal, J., & Nusslein-Volhard, C. (1996). Genes controlling and mediating locomotion behavior of the zebrafish embryo and larva [Number: 1]. *Development*, *123*(1), 399–413. https://doi.org/10.1242/dev.123.1.399

Graving, J. M., Chae, D., Naik, H., Li, L., Koger, B., Costelloe, B. R., & Couzin, I. D. (2019). DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning (I. T. Baldwin, J. W. Shaevitz, J. W. Shaevitz, & G. Stephens, Eds.) [Publisher: eLife Sciences Publications, Ltd]. *eLife*, *8*, e47994. https://doi.org/10.7554/eLife.47994

Greener, J. G., Kandathil, S. M., Moffat, L., & Jones, D. T. (2022). A guide to machine learning for biologists [Number: 1]. *Nature Reviews Molecular Cell Biology*, *23*(1), 40–55. https://doi.org/10.1038/s41580-021-00407-0

Haas, A., Rossberg, A., Schuff, D. L., Titzer, B. L., Holman, M., Gohman, D., Wagner, L., Zakai, A., & Bastien, J. (2017). Bringing the web up to speed with WebAssembly. *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 185–200. https://doi.org/10.1145/3062341.3062363

Hernandez, R. E., Galitan, L., Cameron, J., Goodwin, N., & Ramakrishnan, L. (2018). Delay of Initial Feeding of Zebrafish Larvae Until 8 Days Postfertilization Has No Impact on Survival or Growth Through the Juvenile Stage [Number: 5]. *Zebrafish*, *15*(5), 515–518. https://doi.org/10.1089/zeb.2018.1579

Holtzman, N. G., Iovine, M. K., Liang, J. O., & Morris, J. (2016). Learning to Fish with Genetics: A Primer on the Vertebrate Model Danio rerio [Number: 3]. *Genetics*, *203*(3), 1069–1089. https://doi.org/10.1534/genetics.116.190843

Hotz, A. L., Jamali, A., Rieser, N. N., Niklaus, S., Aydin, E., Myren-Svelstad, S., Lalla, L., Jurisch-Yaksi, N., Yaksi, E., & Neuhauss, S. C. F. (2022). Loss of glutamate transporter eaat2a leads to aberrant neuronal excitability, recurrent epileptic seizures, and basal hypoactivity [Number: 1]. *Glia*, *70*(1), 196–214. https://doi.org/10.1002/glia.24106

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3 [arXiv: 1905.02244]. *arXiv:1905.02244 [cs]*. http://arxiv.org/abs/1905.02244

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications [arXiv: 1704.04861]. *arXiv:1704.04861 [cs]*. http://arxiv.org/abs/1704.04861

Howe, K., Clark, M. D., Torroja, C. F., Torrance, J., Berthelot, C., Muffato, M., Collins, J. E., Humphray, S., McLaren, K., Matthews, L., McLaren, S., Sealy, I., Caccamo, M., Churcher, C., Scott, C., Barrett, J. C., Koch, R., Rauch, G.-J., White, S., … Stemple, D. L. (2013). The zebrafish reference genome sequence and its relationship to the human genome [Number: 7446]. *Nature*, *496*(7446), 498–503. https://doi.org/10.1038/nature12111

Ioffe, S., & Szegedy, C. (2015, March). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Issue: arXiv:1502.03167 arXiv: 1502.03167 [cs]]. https://doi.org/10.48550/arXiv.1502.03167

Joo, W., Vivian, M. D., Graham, B. J., Soucy, E. R., & Thyme, S. B. (2020). A Customizable Low-Cost System for Massively Parallel Zebrafish Behavioral Phenotyping. *Frontiers in Behavioral Neuroscience*, *14*, 606900. https://doi.org/10.3389/fnbeh.2020.606900

Kalueff, A. V., Gebhardt, M., Stewart, A. M., Cachat, J. M., Brimmer, M., Chawla, J. S., Craddock, C., Kyzar, E. J., Roth, A., Landsman, S., Gaikwad, S., Robinson, K., Baatrup, E., Tierney, K., Shamchuk, A., Norton, W., Miller, N., Nicolson, T., Braubach, O., … Zebrafish Neuroscience Research Consortium. (2013). Towards a comprehensive catalog of zebrafish behavior 1.0 and beyond [Number: 1]. *Zebrafish*, *10*(1), 70–86. https://doi.org/10.1089/zeb.2012.0861

Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization [arXiv: 1412.6980]. *arXiv:1412.6980 [cs]*. http://arxiv.org/abs/1412.6980

Lee, H. B., Schwab, T. L., Sigafoos, A. N., Gauerke, J. L., Krug, R. G., Serres, M. R., Jacobs, D. C., Cotter, R. P., Das, B., Petersen, M. O., Daby, C. L., Urban, R. M., Berry, B. C., & Clark, K. J. (2019). Novel zebrafish behavioral assay to identify modifiers of the rapid, nongenomic stress response [Number: 2]. *Genes, Brain, and Behavior*, *18*(2), e12549. https://doi.org/10.1111/gbb.12549

Liu, K. S., & Fetcho, J. R. (1999). Laser Ablations Reveal Functional Relationships of Segmental Hindbrain Neurons in Zebrafish [Number: 2]. *Neuron*, *23*(2), 325–335. https://doi.org/10.1016/S0896-6273(00)80783-7

Liu, Y., Carmer, R., Zhang, G., Venkatraman, P., Brown, S. A., Pang, C.-P., Zhang, M., Ma, P., & Leung, Y. F. (2015). Statistical Analysis of Zebrafish Locomotor Response [Number: 10]. *PloS One*, *10*(10), e0139521. https://doi.org/10.1371/journal.pone.0139521

Lucore, E. C., & Connaughton, V. P. (2021). Observational learning and irreversible starvation in first-feeding zebrafish larvae: Is it okay to copy from your friends? *Zoology*, *145*, 125896. https://doi.org/10.1016/j.zool.2021.125896

MacRae, C. A., & Peterson, R. T. (2023). Zebrafish as a Mainstream Model for In Vivo Systems Pharmacology and Toxicology [Number: 1]. *Annual Review of Pharmacology and Toxicology*, *63*(1), 43–64. https://doi.org/10.1146/annurev-pharmtox-051421-105617

Macrì, S., Neri, D., Ruberto, T., Mwaffo, V., Butail, S., & Porfiri, M. (2017). Three-dimensional scoring of zebrafish behavior unveils biological phenomena hidden by two-dimensional analyses [Number: 1]. *Scientific Reports*, *7*(1), 1962. https://doi.org/10.1038/s41598-017-01990-z

Marques, J. C., Lackner, S., Félix, R., & Orger, M. B. (2018). Structure of the Zebrafish Locomotor Repertoire Revealed with Unsupervised Behavioral Clustering [Number: 2]. *Current biology: CB*, *28*(2), 181–195.e5. https://doi.org/10.1016/j.cub.2017.12.002

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning [Number: 9]. *Nature Neuroscience*, *21*(9), 1281–1289. https://doi.org/10.1038/s41593-018-0209-y

Maximino, C., Puty, B., Matos Oliveira, K. R., & Herculano, A. M. (2013). Behavioral and neurochemical changes in the zebrafish leopard strain [Number: 5]. *Genes, Brain, and Behavior*, *12*(5), 576–582. https://doi.org/10.1111/gbb.12047

McKeown, K. A., Moreno, R., Hall, V. L., Ribera, A. B., & Downes, G. B. (2012). Disruption of Eaat2b, a glutamate transporter, results in abnormal motor behaviors in developing zebrafish [Number: 2]. *Developmental Biology*, *362*(2), 162–171. https://doi.org/10.1016/j.ydbio.2011.11.001

Mirat, O., Sternberg, J. R., Severi, K. E., & Wyart, C. (2013). ZebraZoom: An automated program for high-throughput behavioral analysis and categorization. *Frontiers in Neural Circuits*, *7*, 107. https://doi.org/10.3389/fncir.2013.00107

Newell, A., Yang, K., & Deng, J. (2016, July). Stacked Hourglass Networks for Human Pose Estimation [Issue: arXiv:1603.06937 arXiv: 1603.06937 [cs]]. https://doi.org/10.48550/arXiv.1603.06937

Nilsson, S. R., Goodwin, N. L., Choong, J. J., Hwang, S., Wright, H. R., Norville, Z. C., Tong, X., Lin, D., Bentzley, B. S., Eshel, N., McLaughlin, R. J., & Golden, S. A. (2020, April). *Simple Behavioral Analysis (SimBA) – an open source toolkit for computer classification of complex social behaviors in experimental animals* (preprint). Animal Behavior and Cognition. https://doi.org/10.1101/2020.04.19.049452

Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S.-H., Murthy, M., & Shaevitz, J. W. (2019). Fast animal pose estimation using deep neural networks [Number: 1]. *Nature Methods*, *16*(1), 117–125. https://doi.org/10.1038/s41592-018-0234-5

Perkel, J. M. (2024). No installation required: How WebAssembly is changing scientific computing [Bandiera_abtest: a Cg_type: Technology Feature Publisher: Nature Publishing Group Subject_term: Technology, Computational biology and bioinformatics, Computer science, Software, Education]. *Nature*, *627*(8003), 455–456. https://doi.org/10.1038/d41586-024-00725-1

R Core Team. (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Randlett, O., Haesemeyer, M., Forkin, G., Shoenhard, H., Schier, A. F., Engert, F., & Granato, M. (2019). Distributed Plasticity Drives Visual Habituation Learning in Larval Zebrafish [Number: 8]. *Current Biology*, *29*(8), 1337–1345.e4. https://doi.org/10.1016/j.cub.2019.02.039

Ravan, A., Feng, R., Gruebele, M., & Chemla, Y. R. (2023). Rapid automated 3-D pose estimation of larval zebrafish using a physical model-trained neural network. *PLoS computational biology*, *19*(10), e1011566. https://doi.org/10.1371/journal.pcbi.1011566

Romero-Ferrero, F., Bergomi, M. G., Hinz, R. C., Heras, F. J. H., & de Polavieja, G. G. (2019). Idtracker.ai: Tracking all individuals in small or large collectives of unmarked animals [Number: 2 Publisher: Nature Publishing Group]. *Nature Methods*, *16*(2), 179–182. https://doi.org/10.1038/s41592-018-0295-5

Ronneberger, O., Fischer, P., & Brox, T. (2015, May). U-Net: Convolutional Networks for Biomedical Image Segmentation [Issue: arXiv:1505.04597 arXiv: 1505.04597 [cs]]. https://doi.org/10.48550/arXiv.1505.04597

Sakai, C., Ijaz, S., & Hoffman, E. J. (2018). Zebrafish Models of Neurodevelopmental Disorders: Past, Present, and Future. *Frontiers in Molecular Neuroscience*, *11*, 294. https://doi.org/10.3389/fnmol.2018.00294

Saleh, A., Jones, D., Jerry, D., & Azghadi, M. R. (2023). MFLD-net: A lightweight deep learning network for fish morphometry using landmark detection. *Aquatic Ecology*, *57*(4), 913–931. https://doi.org/10.1007/s10452-023-10044-8

Sandler, M., Baccash, J., Zhmoginov, A., & Howard, A. (2019, October). Non-discriminative data or weak model? On the relative importance of data and model resolution [Issue: arXiv:1909.03205 arXiv: 1909.03205 [cs]]. https://doi.org/10.48550/arXiv.1909.03205

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2019). MobileNetV2: Inverted Residuals and Linear Bottlenecks [arXiv: 1801.04381]. *arXiv:1801.04381 [cs]*. http://arxiv.org/abs/1801.04381

Schwartz, A. V., Sant, K. E., Navarrete, J., & George, U. Z. (2021). Mathematical modeling of the interaction between yolk utilization and fish growth in zebrafish, Danio rerio [Number: 9]. *Development (Cambridge, England)*, *148*(9), dev193508. https://doi.org/10.1242/dev.193508

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning [Number: 1]. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Stednitz, S. J., & Washbourne, P. (2020). Rapid Progressive Social Development of Zebrafish [Number: 1]. *Zebrafish*, *17*(1), 11–17. https://doi.org/10.1089/zeb.2019.1815

Sztal, T. E., & Stainier, D. Y. R. (2020). Transcriptional adaptation: A mechanism underlying genetic robustness [Number: 15]. *Development (Cambridge, England)*, *147*(15), dev186452. https://doi.org/10.1242/dev.186452

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2017, November). Instance Normalization: The Missing Ingredient for Fast Stylization [arXiv: 1607.08022 [cs]]. https://doi.org/10.48550/arXiv.1607.08022

Wickham, H. (2016). *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org

Wu, Y., & He, K. (2018, June). Group Normalization [Issue: arXiv:1803.08494 arXiv: 1803.08494 [cs]]. https://doi.org/10.48550/arXiv.1803.08494