zea: A Toolbox for Cognitive Ultrasound Imaging

Tristan S. W. Stevens[1¶], **Wessel L. van Nierop**[1], **Ben Luijten**[1], **Vincent van de Schaft**[1], **Oisín Nolan**[1], **Beatrice Federici**[1], **Louis D. van Harten**[1], **Simon W. Penninga**[1], **Noortje I. P. Schueler**[1], and **Ruud J. G. van Sloun**[1]

1 Eindhoven University of Technology, the Netherlands ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 25 June 2025
Published: unpublished

Summary

Ultrasound imaging is a powerful medical imaging modality that is widely used in clinical settings for various applications, including obstetrics, cardiology, and abdominal imaging. While ultrasound imaging is non-invasive, real-time, and relatively low-cost compared to other imaging modalities such as MRI or CT, it still faces challenges in terms of image quality and interpretation. Many signal processing steps are required to extract useful information from the raw ultrasound data, such as filtering, beamforming, and image reconstruction. Traditional ultrasound imaging techniques often suffer from reduced image quality as naive assumptions are made in these processing steps, which do not account for the complex nature of ultrasound signals. Furthermore, acquisition (action) and reconstruction (perception) of ultrasound is often performed disjointly. Cognitive ultrasound imaging (Sloun, 2024), see Figure 1, is a novel approach that aims to address these challenges by leveraging more powerful generative models, enabled by advances in deep learning, to close the action-perception loop. This approach requires a redesign of current common ultrasound imaging pipeline, where parameters are expected to be changed dynamically based on past and current observations. Furthermore, the high-dimensional nature of ultrasound data requires powerful deep generative models to learn the structured distribution of ultrasound signals and to effectively solve inverse problems that capture the challenges of ultrasound imaging (T. S. Stevens et al., 2025). This necessitates a flexible and efficient toolbox that can handle the complexities of cognitive ultrasound imaging, including a real-time ultrasound reconstruction pipeline, dynamic parameter adjustment, and advanced generative modeling.

We present zea (pronounced *ze-yah*), a Python package for cognitive ultrasound imaging that provides a flexible, modular and differentiable pipeline for ultrasound data processing, as well as a collection of pre-defined models for ultrasound image and signal processing. The toolbox is designed to be easy to use, with a high-level interface that allows users to define their own ultrasound reconstruction pipelines, and to integrate deep learning models into the pipeline. The toolbox is built on top of Keras 3 (Chollet & others, 2015), which provides a framework for building and training deep learning models with the three major deep learning frameworks as backend: TensorFlow (Abadi et al., 2016), PyTorch (Paszke et al., 2019) and JAX (Bradbury et al., 2018). This means that it is easy to integrate a custom ultrasound reconstruction pipeline in a machine learning workflow. In the past few years, several works have used and contributed to zea, including Luijten et al. (2020), Van de Schaft et al. (2025), T. S. W. Stevens et al. (2024), Nolan et al. (2025), Federici et al. (2024), T. S. W. Stevens, Nolan, Robert, et al. (2025), Penninga et al. (2025) and T. S. W. Stevens, Nolan, Somphone, et al. (2025).

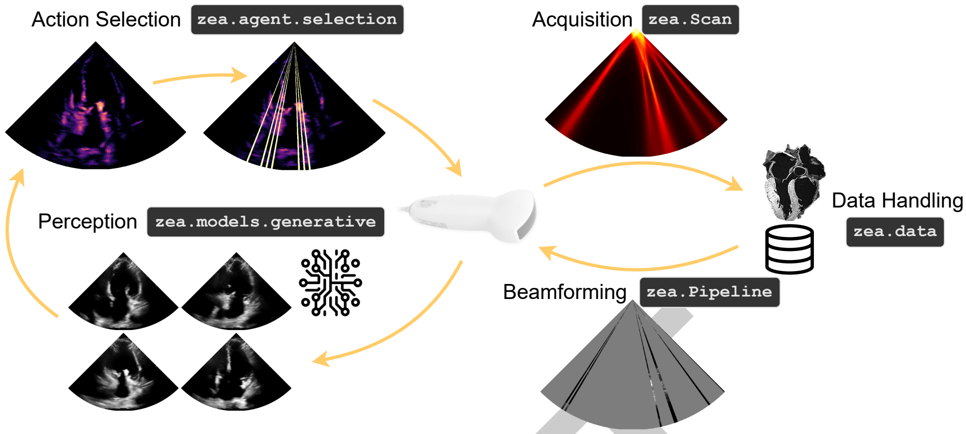

Figure 1: High-level overview of an ultrasound perception-action loop implemented in zea.

Statement of need

The ultrasound research community has advanced significantly due to publically available high-quality software, including simulation tools such as `Field II` (Jensen, 2004) and `k-wave` (Treeby & Cox, 2010), as well as reconstruction and real-time processing libraries like `USTB` (Rodriguez-Molares et al., 2017), `MUST` (Garcia, 2021), `ARRUS` (Jarosik & others, 2020), `FAST` (Smistad, 2021), `QUPS` (Brevett, 2024), and `vbeam` (Kvalevåg et al., 2023). However, existing solutions are not well equipped for cognitive ultrasound imaging, where the integration of deep learning and dynamic, closed-loop ultrasound reconstruction pipelines is essential. Our aim with zea is to provide a complementary, highly flexible and differentiable pipeline written in a modern deep learning framework, as well as offer a convenient platform for pretrained models. This addresses the need for a modular and extensible library that supports cognitive ultrasound workflows and seamless integration with state-of-the-art machine learning models. While the full realization of cognitive ultrasound imaging remains an ongoing effort, we hope this toolbox will help spur further research and development in the field.

Overview of functionality

zea is an open-source Python package, available at http://github.com/tue-bmd/zea, that consists of the following core components:

- **Data**: A set of data handling classes such as `zea.data.File`, `zea.data.Dataset` and `make_dataloader()`, suited for machine learning workflows. zea works with HDF5 files, storing data and acquisition parameters together in a single file. Additionally, we provide examples and conversion scripts for popular ultrasound datasets, such as CAMUS (Leclerc et al., 2019), PICMUS (Liebgott et al., 2016), and EchoNet (Ouyang et al., 2019).
- **Pipeline**: A modular and differentiable pipeline class that allows users to define a sequence of operations (`zea.Operation`) to process ultrasound data. The pipeline is stateless and supports *Just in Time* (JIT) compilation. Ultimately, this allows for dynamic parameter adjustment, as well as real-time integration of deep learning models inside the ultrasound reconstruction pipeline.
- **Models**: A collection of pre-defined models for ultrasound image and signal processing. Similar to the data, these models can be loaded locally or from the Hugging Face Hub. Besides supervised models, zea also provides a set of (deep) generative models, with an interface to solve inverse problems in ultrasound imaging within a probabilistic machine learning framework.

- **Agents**: A set of tools to interact with the pipeline and models. These agents can be used to alter the pipeline parameters, or select a subset of acquired data. The agent module closes the action-perception loop, tying together acquisition and reconstruction of ultrasound data.

Example usage

Below, we will show a brief overview of how to use the main components of zea, including the data handling, pipeline, models, and agents. For more detailed examples and use cases, please refer to the example notebooks available on the documentation: https://zea.readthedocs.io/.

Data

zea stores data as well as acquisition parameters together in HDF5 files, which can be easily loaded and saved through the zea.data API.

```python
import zea
# path to a local or remote HDF5 file in zea format
path = "hf://zeahub/..."

# read data and acquisition parameters from an HDF5 file
with zea.File(path, mode="r") as file:
    file.summary()

    data = file.load_data("raw_data", indices=[0]) # load first frame
    scan = file.scan()
    probe = file.probe()
```

Using zea.File to load individual data files or zea.Dataset to manage multiple files is convenient for rapid prototyping and exploration. However, more demanding workflows, such as training deep learning models, benefit from more robust data loading utilities. To address this, zea offers the make_dataloader() function, which is fully compatible with zea formatted HDF5 files. This utility streamlines the preparation of data for training by supporting essential features like batching, shuffling, caching, and preprocessing.

```python
from zea.backend.tensorflow import make_dataloader

dataset_path = "hf://zeahub/camus-sample/val"
dataloader = make_dataloader(
    dataset_path,
    key="data/image_sc",
    batch_size=4,
    shuffle=True,
    clip_image_range=[-60, 0],
    image_range=[-60, 0],
    normalization_range=[0, 1],
    image_size=(256, 256),
    resize_type="resize", # or "center_crop or "random_crop"
    seed=4,
)

for batch in dataloader:
    ... # your training loop here
```

Pipeline

The core of zea is a modular and differentiable pipeline class designed for ultrasound data processing. Built on modern deep learning frameworks, this pipeline enables users to compose both built-in and custom operations derived from the base class `zea.Operation`, e.g. `DelayAndSum`, including the integration of deep learning models within the processing workflow. The pipeline is stateless, meaning it does not retain information between operations, which facilitates dynamic parameter adjustment and supports real-time reconstruction scenarios. Additionally, the pipeline offers *Just-in-Time* (JIT) compilation, which can significantly enhance performance by optimizing the execution of operations at runtime.

```python
import zea
from zea.ops import *

pipeline = zea.Pipeline(
    operations=[
        Demodulate(),                 # IQ demodulation
        PatchedGrid(                  # Memory efficient processing
            operations=[
                TOFCorrection(),      # Time-of-flight correction
                PfieldWeighting(),    # Weighting by estimated pressure field
                DelayAndSum(),        # Weighted sum
            ],
            num_patches=100,
        ),
        EnvelopeDetect(),             # Envelope detection
        Normalize(),                  # Normalization
        LogCompress(),                # to dB scale (B-mode)
    ],
)

# local or remote Hugging Face path to hdf5 file
path = (
    "hf://zeahub/picmus/database/experiments/contrast_speckle/"
    "contrast_speckle_expe_dataset_rf/contrast_speckle_expe_dataset_rf.hdf5"
)
data, scan, probe = zea.load_file(
    path=path,
    data_type="raw_data",
    scan_kwargs={"xlims": (-20e-3, 20e-3), "zlims": (0e-3, 80e-3)},
)

# place parameters on e.g. GPU
parameters = pipeline.prepare_parameters(probe, scan)

inputs = {pipeline.key: data}

# parameters can be dynamically passed here as keyword arguments, e.g., sound_speed
sound_speed = 1540  # m/s
outputs = pipeline(**inputs, **parameters, sound_speed=sound_speed)

image = outputs[pipeline.output_key]
```

Models

One contribution of zea is to extend conventional ultrasound imaging pipelines with data-driven models, such as deep generative models, to learn the structured distribution of ultrasound signals. This allows for more powerful reconstruction and denoising capabilities, as well as the ability to perform inverse problems in a probabilistic machine learning framework. The zea.models subpackage provides a collection of pre-defined models for ultrasound image and signal processing, which can be easily integrated into the reconstruction pipeline.

```python
import keras
import zea
from zea.models.diffusion import DiffusionModel

# use a built-in preset or a local / remote HF path to your model
model = DiffusionModel.from_preset("diffusion-echonet-dynamic")

# sample from the model's prior distribution
prior_samples = model.sample(n_samples=16, n_steps=90, verbose=True)
prior_samples = keras.ops.squeeze(prior_samples, axis=-1)

# set up a pipeline to process the prior samples into images
pipeline = zea.Pipeline([zea.ops.ScanConvert(order=2)])

parameters = {
    "theta_range": [-0.78, 0.78],  # [-45, 45] in radians
    "rho_range": [0, 1],
}
parameters = pipeline.prepare_parameters(**parameters)

# process the prior samples through the pipeline
images = pipeline(data=prior_samples, **parameters)["data"]

# visualize
fig, _ = zea.visualize.plot_image_grid(images, vmin=-1, vmax=1)
```

Which will generate the samples as seen in Figure 2.

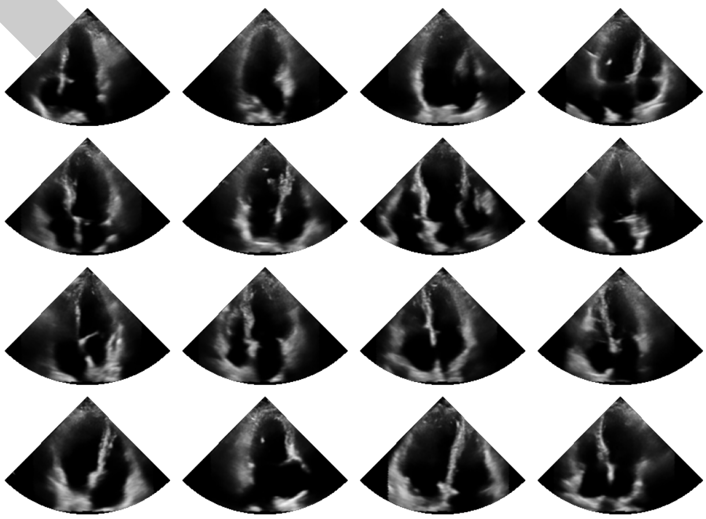

Figure 2: Diffusion posterior samples of a model trained on the EchoNet-dynamic dataset.

Agent

The agent subpackage provides tools and utilities for agent-based algorithms within the zea framework. These agents consist of tools that can alter pipeline or model parameters, select a subset of acquired data, or perform other actions that are necessary to close the action-perception loop in cognitive ultrasound imaging. Currently, it supports intelligent focused transmit scheme design via *active perception* (Sloun, 2024), with implementations of key algorithms such as *Greedy Entropy Minimization*, and mask generation functions to create measurement models mapping from fully-observed to subsampled data.

```python
import keras
import zea

# (batch, height, width)
data = ...

# create a Greedy Entropy Minimization agent
agent = zea.agent.selection.GreedyEntropy(
    n_actions=width // 8,
    n_possible_actions=width,
    img_width=width,
    img_height=height,
)

# these would normally be sampled from a posterior distribution p(x ¦ y)
particles = keras.random.uniform((batch_size, 10, height, width))
lines, mask = agent.sample(particles)

measurement = keras.ops.where(mask, data, min_val)
images = keras.ops.concatenate([data, measurement], axis=0)
scanconvert = zea.ops.ScanConvert(order=2)
images = scanconvert(
  data=images, rho_range=[0, 1], theta_range=[-0.78, 0.78]
)["data"]

# visualize
vmin, vmax = dynamic_range
fig, _ = zea.visualize.plot_image_grid(
  images, vmin=vmin, vmax=vmax, cmap="gray"
)
```

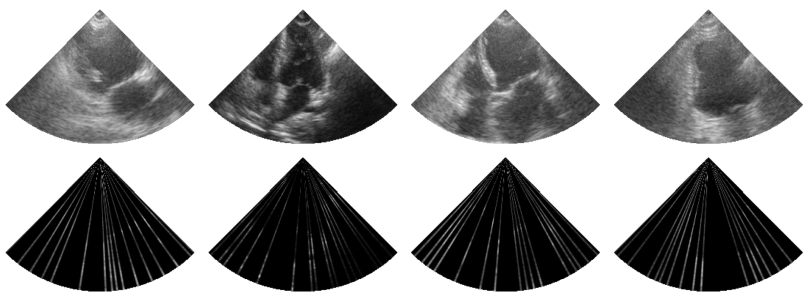

Figure 3: Selected scan-lines as chosen by a Greedy Entropy Minimization agent.

Availability, Development, and Documentation

zea is available through PyPI via `pip install zea`, and the development version is available via GitHub. GitHub Actions manage continuous integration through automated code testing (PyTest), code linting and formatting (Ruff), and documentation generation (Sphinx). The documentation is hosted on ReadTheDocs. At the time of writing, 15 example notebooks are available, covering the various discussed components of the toolbox. The package is licensed under the Apache License 2.0, which allows for both academic and commercial use.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P. A., Vasudevan, V., Warden, P., … Zheng, X. (2016). TensorFlow: A System for Large-scale Machine Learning. In K. Keeton & T. Roscoe (Eds.), *12th USENIX symposium on operating systems design and implementation, OSDI 2016, savannah, GA, USA, november 2-4, 2016* (pp. 265–283). USENIX Association. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). *JAX: Composable transformations of Python+NumPy programs* (Version 0.3.13). http://github.com/jax-ml/jax

Brevett, T. (2024). QUPS: A MATLAB Toolbox for Rapid Prototyping of Ultrasound Beamforming and Imaging Techniques. *J. Open Source Softw.*, *9*(101), 6772. https://doi.org/10.21105/JOSS.06772

Chollet, F., & others. (2015). *Keras*. https://keras.io.

Federici, B., Sloun, R. J. G. van, & Mischi, M. (2024). Active Inference for Closed-loop transmit beamsteering in Fetal Doppler Ultrasound. *CoRR*, *abs/2410.04869*. https://doi.org/10.48550/ARXIV.2410.04869

Garcia, D. (2021). Make the most of MUST, an open-source Matlab UltraSound Toolbox. *2021 IEEE International Ultrasonics Symposium (IUS)*, 1–4. https://doi.org/10.1109/IUS52206.2021.9593605

Jarosik, P., & others. (2020). *ARRUS: Research/remote ultrasound platform*. https://github.com/us4useu/arrus

Jensen, J. A. (2004). Simulation of advanced ultrasound systems using Field II. *2004 2nd IEEE International Symposium on Biomedical Imaging: Nano to Macro (IEEE Cat No. 04EX821)*, 636–639 Vol. 1. https://doi.org/10.1109/ISBI.2004.1398618

Kvalevåg, M. D., Vrålstad, A. E., Rindal, O. M. H., Bjåstad, T. G., Denarie, B., Kristoffersen, K., Måsøy, S.-E., & Løvstakken, L. (2023). Vbeam: A Fast and Differentiable Beamformer for Optimizing Ultrasound Imaging. *2023 IEEE International Ultrasonics Symposium (IUS)*, 1–4. https://doi.org/10.1109/IUS51837.2023.10307255

Leclerc, S., Smistad, E., Pedrosa, J., Østvik, A., Cervenansky, F., Espinosa, F., Espeland, T., Berg, E. A. R., Jodoin, P.-M., Grenier, T., Lartizien, C., D'hooge, J., Løvstakken, L., & Bernard, O. (2019). Deep Learning for Segmentation Using an Open Large-scale Dataset in 2D Echocardiography. *IEEE Trans. Medical Imaging*, *38*(9), 2198–2210. https://doi.org/10.1109/TMI.2019.2900516

Liebgott, H., Rodriguez-Molares, A., Cervenansky, F., Jensen, J. A., & Bernard, O. (2016). Plane-wave Imaging Challenge in Medical Ultrasound. *2016 IEEE International Ultrasonics Symposium (IUS)*, 1–4. https://doi.org/10.1109/ULTSYM.2016.7728908

Luijten, B., Cohen, R., Bruijn, F. J. de, Schmeitz, H. A. W., Mischi, M., Eldar, Y. C., & Sloun, R. J. G. van. (2020). Adaptive Ultrasound Beamforming Using Deep Learning. *IEEE Trans. Medical Imaging*, *39*(12), 3967–3978. https://doi.org/10.1109/TMI.2020.3008537

Nolan, O., Stevens, T. S. W., Nierop, W. L. van, & Sloun, R. van. (2025). Active Diffusion Subsampling. *Trans. Mach. Learn. Res.*, *2025*. https://openreview.net/forum?id=OGifiton47

Ouyang, D., He, B., Ghorbani, A., Lungren, M. P., Ashley, E. A., Liang, D. H., & Zou, J. Y. (2019). Echonet-dynamic: A large new cardiac motion video data resource for medical machine learning. *NeurIPS ML4H Workshop: Vancouver, BC, Canada*, *5*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An Imperative Style, High-performance Deep Learning Library. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, NeurIPS 2019, december 8-14, 2019, vancouver, BC, canada* (pp. 8024–8035). https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

Penninga, S. W., Gorp, H. V., & Sloun, R. J. G. van. (2025). Deep Sylvester Posterior Inference for Adaptive Compressed Sensing in Ultrasound Imaging. *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*, 1–5. https://doi.org/10.1109/ICASSP49660.2025.10888253

Rodriguez-Molares, A., Rindal, O. M. H., Bernard, O., Nair, A., Lediju Bell, M. A., Liebgott, H., Austeng, A., & LØvstakken, L. (2017). The UltraSound ToolBox. *2017 IEEE International Ultrasonics Symposium (IUS)*, 1–4. https://doi.org/10.1109/ULTSYM.2017.8092389

Sloun, R. J. G. van. (2024). Active inference and deep generative modeling for cognitive ultrasound. *CoRR*, *abs/2410.13310*. https://doi.org/10.48550/ARXIV.2410.13310

Smistad, E. (2021). FAST: A framework for high-performance medical image computing and visualization. In S. McIntosh-Smith (Ed.), *IWOCL'21: International workshop on OpenCL, munich germany, april, 2021* (pp. 14:1–14:2). ACM. https://doi.org/10.1145/3456669.3456717

Stevens, T. S. W., Meral, F. C., Yu, J., Apostolakis, I. Z., Robert, J.-L., & Sloun, R. J. G. van. (2024). Dehazing Ultrasound Using Diffusion Models. *IEEE Trans. Medical Imaging*, *43*(10), 3546–3558. https://doi.org/10.1109/TMI.2024.3363460

Stevens, T. S. W., Nolan, O., Robert, J.-L., & Sloun, R. J. G. van. (2025). Sequential Posterior Sampling with Diffusion Models. *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*, 1–5. https://doi.org/10.1109/ICASSP49660.2025.10889752

Stevens, T. S. W., Nolan, O., Somphone, O., Robert, J.-L., & Sloun, R. J. G. van. (2025). High Volume Rate 3D Ultrasound Reconstruction with Diffusion Models. *CoRR*, *abs/2505.22090*. https://doi.org/10.48550/ARXIV.2505.22090

Stevens, T. S., Overdevest, J., Nolan, O., Nierop, W. L. van, Sloun, R. J. van, & Eldar, Y. C. (2025). Deep generative models for Bayesian inference on high-rate sensor data: Applications in automotive radar and medical imaging. *Philosophical Transactions A*, *383*(2299), 20240327. https://doi.org/10.1098/rsta.2024.0327

Treeby, B. E., & Cox, B. T. (2010). K-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields. *Journal of Biomedical Optics*, *15*(2), 021314–021314. https://doi.org/10.1117/1.3360308

Van de Schaft, V., Nolan, O., & Van Sloun, R. J. (2025). Off-grid Ultrasound Imaging by Stochastic Optimization. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 1–1. https://doi.org/10.1109/TUFFC.2025.3586377