Supplementary Material

Cloth in the Wind: A Case Study of Physical Measurement through Simulation

1. Table of Contents

The supplementary material has the following content:

- **Section 2** and **Algorithm 1**: Formalization of the batched version of the spectral decomposition layer.
- **Section 3** and **Figure 1**: Details on the collection of real wind speed measurements and video recordings.
- **Section 4**: Details on the video data augmentation and optimization for the training of all networks.
- **Section 5** and **Figure 4**: Additional experiments on refined measurements for the hanging cloth dataset and details on our ClothSim dataset.
- **Table 1**: Supplement to Table 2 in the main paper with wind speed regression results on our FlagSim dataset.
- **Figure 3**: Supplement to Figure 8 in the main paper to include more refined measurement examples.
- **Listing 1**, **Listing 2** and **Listing 3**: Specification of ArcSim scene and material configuration files for flag and cloth simulations.
- **Table 2**: Exhaustive list of Blender's rendering parameters for generating the FlagSim and ClothSim datasets.

2. Spectral Decomposition Layer

To support Section 4.3 in the main paper, we formalize the batched version of the spectral decomposition layer in Algorithm 1. Given a batch of video clips as input, our spectral layer applies the discrete Fourier transform along the temporal dimension to compute the temporal frequency spectrum. From the periodogram, we can select the top-k strongest frequency responses and their corresponding spectral power. The resulting frequency maps and power maps all have the same dimensions and can, therefore, be stacked as a multi-channel image. These tensors can be further processed by standard 2D convolution layers to learn frequency-based feature representations. The proposed layer is efficiently implemented in PyTorch [6] to run on the GPU using the `torch.irfft` operation. The source code is available through the project website.

Algorithm 1 Spectral Decomposition Layer

Input. Video tensor $\boldsymbol{x}$ of shape $[N_b, C, N_t, H, W]$
Input. Number of frequency peaks to select, k
Output. Decomposition of shape $[N_b, 2kC, H, W]$

procedure SpectralDecompositionLayer($\boldsymbol{x}$)
    Reshape $\boldsymbol{x}$ to $[N_b CHW, N_t]$ to obtain batch of signals
    Apply a Hanning window to signals
    Compute the DFT of signals using Eq. 4 (main paper)
    Compute periodogram of signals $I(\omega)$
    Select top-k peaks of $I(\omega)$ and corresponding ω's
    $P \leftarrow$ top-k peaks of $I(\omega)$ reshaped to $[N_b, kC, H, W]$
    $\Omega \leftarrow$ corresponding ω's reshaped to $[N_b, kC, H, W]$
    return P, Ω
end procedure

3. Real-World Flag Dataset Acquisition

We here describe our data acquisition to obtain real-world wind speed measurements serving as ground-truth for our final experiment. To accurately gauge the wind speed next to the flag, we have obtained two anemometers:

- SkyWatch BL-400: windmill-type anemometer
- Testo 410i: vane-type anemometer

The measurement accuracy of both anemometers is 0.2 m s^{-1}. To verify the correctness of both anemometers, we have checked that both wind meters report the same wind speeds before usage. After that, we use the SkyWatch BL-400 anemometer for our measurements as it measures omnidirectional which is more convenient. We hoisted the anemometer in a flag pole such that the wind speeds are measured at the same height as the flag. Wind speed measurements are recorded at 1 second intervals and interfaced to the computer. In Figure 1 we display an example measurement and report the dataset's wind speed distribution. For the experiments (Section 6, main paper, last section), we randomly sample video clips of 30 consecutive frames from our video recordings and consider the ground-truth wind speed to be the average over the last minute. This procedure ensures that small wind speed deviations and measurement errors are averaged out over time.

To capture the videos, we use a Panasonic HC-V770 video camera. The camera records at 1920 × 1080 at 60 frames

per second. We perform post-processing of the videos in the following ways. Firstly, we temporally subsample the video frames at 25 fps such that the clips are in accordance with the frame step size in the physics simulator. Moreover, we assert that the video recordings are temporally aligned with the wind speed measurements using their timestamps. Secondly, we manually crop the videos such that the curling flag appears in the approximate center of the frame. After this, the frames are spatially subsampled to 300×300, again in agreement with animations obtained from the render engine.

4. Training Details

Data Augmentation. The examples in the FlagSim dataset are stored as a sequence of 60 JPEG frames of size 300×300. During training, when using less than 60 input frames (30 is used in all experiments), we randomly sample N_t successive frames from each video clip. This is achieved by uniform sampling of a temporal offset within the video. After this, for the sampled sequence of frames, we convert images to grayscale, perform multi-scale random cropping and apply random horizontal flipping [8] to obtain a $N_t \times 1 \times 224 \times 224$ input clip. Finally, we subtract the mean and divide by the standard deviation for each video clip.

Optimization Details. We train all networks using stochastic gradient descent with Adam [4]. We initialize training with a learning rate of 10^{-2} and decay the learning rate with a factor 10 after 20 epochs. To prevent overfitting, we utilize weight decay of $2 \cdot 10^{-3}$ for all networks. Training continues until validation loss plateaus – typically around 40 epochs. Total training time for our spectral decomposition network is about 4 hours on a single Nvidia GeForce GTX Titan X. When training the recurrent models [3, 10] we also perform gradient clipping (max norm of 10) to improve training stability.

5. Experiments on Hanging Cloth Video

Our real-world flag dataset enables us to evaluate our method's measurement performance of external parameters ($v_w \in \boldsymbol{\theta}_e$). However, the cloth's internal parameters are unknown and cannot be evaluated beyond visual inspection. Therefore, we also perform experiments on the hanging cloth dataset of Bouman *et al*. [2]. The authors have carefully determined the internal cloth material properties, which we can leverage for quantitative evaluation of our simulated-refined measurements. Specifically, we assess our method's ability to measure the cloth's *area weight* ($\mathrm{kg\,m^{-2}}$). The method is identical to that explained in the main paper with its results presented in the final experiment of Section 6. However, we retrain the embedding function $s_\phi(\boldsymbol{x})$ on a dataset of hanging cloth simulations, which we refer to as ClothSim. In this section, we will briefly discuss the characteristics of this dataset and report experimental results.

Table 1. External wind speed prediction from simulation. We regress the wind speed ($v_w \in \boldsymbol{\theta}_e$) on our **FlagSim dataset**. The metrics are computed over the 3.5K test examples. Target velocities range from 0 $\mathrm{m\,s^{-1}}$ (no wind) to 10 $\mathrm{m\,s^{-1}}$ (strong wind). Experimental setup is identical to Table 2 in the main paper.

Model	Input Modality	RMSE ↓	Acc@0.5 ↑
Yang *et al*. [10]	$10 \times 227 \times 227$	0.380	0.620
Cardona *et al*. [3]	$30 \times 227 \times 227$	0.271	0.580
ResNet-18	$1 \times 224 \times 224$	0.381	0.615
ResNet-18	$10 \times 224 \times 224$	0.264	0.734
ResNet-18	$20 \times 224 \times 224$	0.207	0.775
SDN (ours)	$20 \times 224 \times 224$	0.183	0.813
SDN (ours)	$30 \times 224 \times 224$	**0.180**	**0.838**

ClothSim Dataset. Following the same procedure as for the FlagSim dataset, we additionally generate a dataset of simulated hanging cloth excited by a constant wind force. The main difference between the FlagSim dataset is the wider variety of cloth material. Specifically, we use all the materials presented in [7] available in ArcSim. The increased diversity allows us to model the dynamics in real-world hanging cloth recording [2]. Our dataset shares similarity with the simulated hanging cloth dataset of [10]. However, in their work, the dataset is employed to train a classifier for predicting the material class. In Listing 2 and Table 2 we present an exhaustive overview of the simulation and render parameters that were used for generating the dataset.

Real-world Parameter Refinement ($\boldsymbol{\theta}_i, \boldsymbol{\theta}_e$). Given the embedding function $s_\phi(\boldsymbol{x})$ trained on ClothSim using contrastive loss, we run our refined measurement experiment on the hanging cloth dataset of Bouman *et al*. [2]. Our goal is to measure the cloth's area weight as we have access to its ground-truth measurement. Unlike for our real-world flag dataset, we do not know the true wind speed beyond the setting of the industrial fan that was used for exciting the fabric artificially. In Figure 4 we report the results for 3 randomly sampled real-world videos.

References

[1] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, 2018. 8

[2] Katherine L Bouman, Bei Xiao, Peter Battaglia, and William T Freeman. Estimating the material properties of fabric from video. In *ICCV*, 2013. 2, 4, 7

[3] Jennifer L Cardona, Michael F Howland, and John O Dabiri. Seeing the wind: Visual wind speed prediction with a coupled convolutional and recurrent neural network. In *NeurIPS*, 2019. 2

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

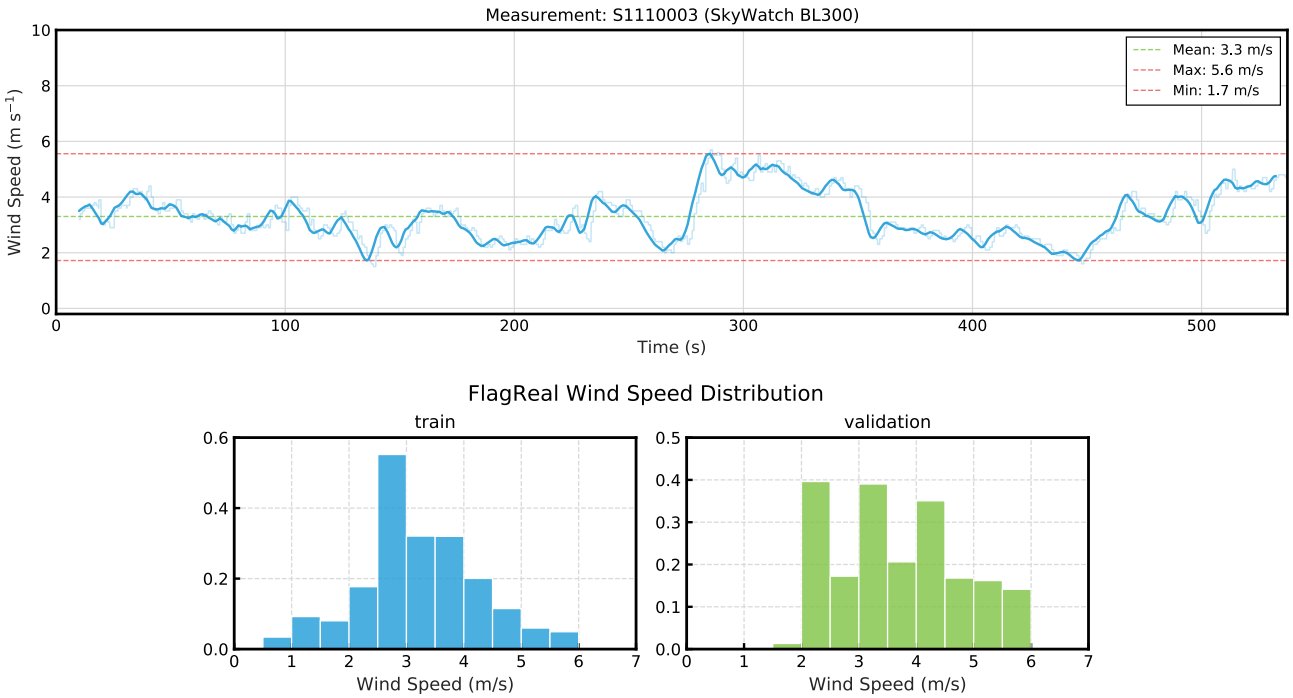

Figure 1. *Top:* Example of the time-varying wind speed as obtained by the SkyWatch BL-400 anemometer positioned directly next to the video-recorded flag. The wind speed is sampled at 1 Hz and interfaced to a computer using bluetooth. For our final experiment, we sample video clips of 30 frames and consider the ground-truth wind speed to be the average wind speed over the last minute. *Bottom:* Distribution statistics of the dataset we collected. Over all 4K non-overlapping videos the average wind speed is 3.2m s^{-1} while the minimum and maximum wind speeds are 0.5m s^{-1} and 6.0m s^{-1} respectively.

[5] Rahul Narain, Armin Samii, and James F O'Brien. Adaptive anisotropic remeshing for cloth simulation. In *SIGGRAPH*, 2012. 6, 8

[6] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 1

[7] Huamin Wang, James F O'Brien, and Ravi Ramamoorthi. Data-driven elastic models for cloth: modeling and measurement. In *SIGGRAPH*, 2011. 2, 8

[8] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015. 2

[9] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 8

[10] Shan Yang, Junbang Liang, and Ming C Lin. Learning-based cloth material recovery from video. In *ICCV*, 2017. 2, 8

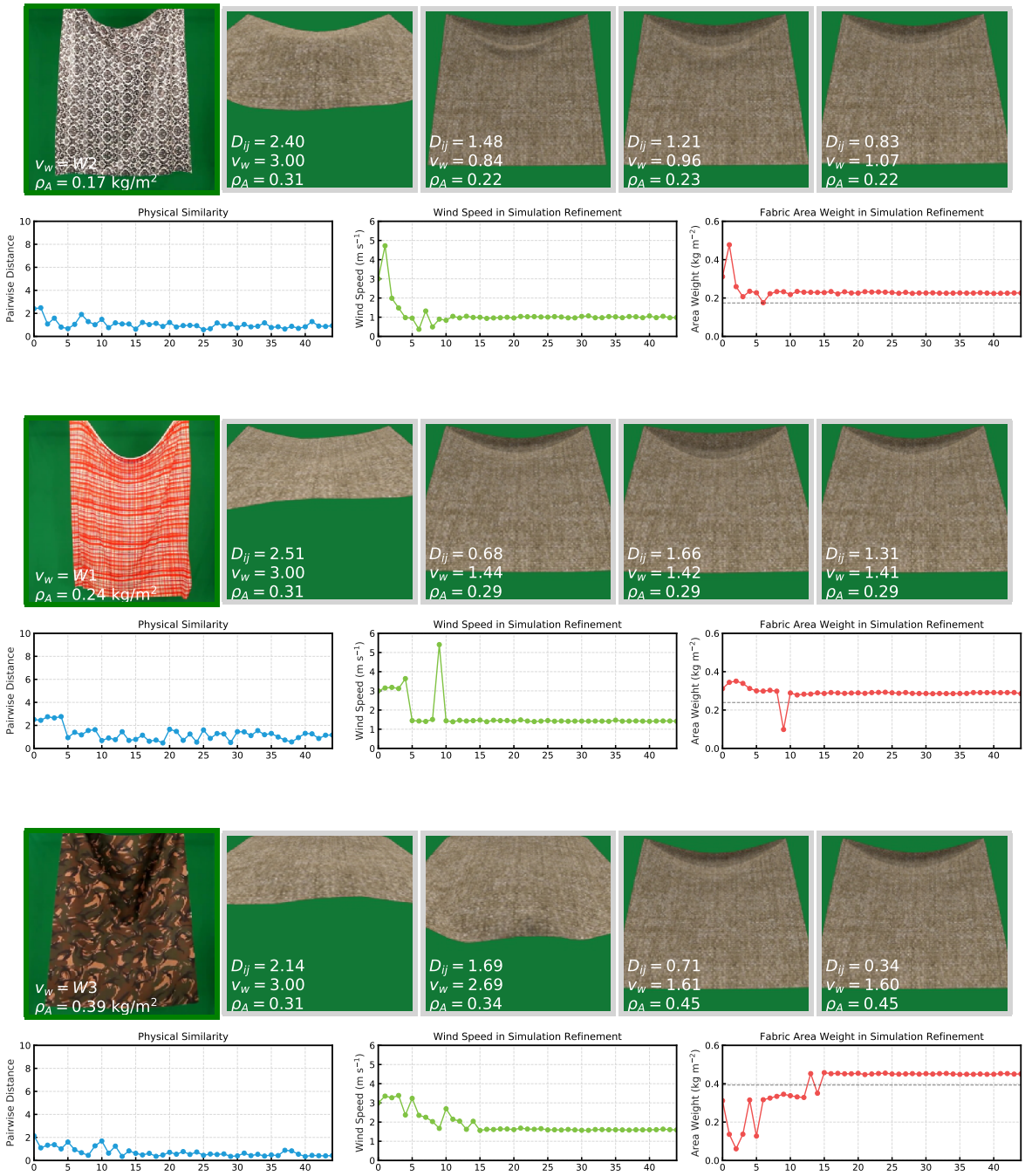

Figure 4. Results for our **hanging cloth refined-measurements** for a random target video capturing the hanging cloth as recorded by Bouman *et al*. [2]. The five images are the center frames of the real-world target video (left) and simulations throughout the refinement process after $t = 0, 10, 20, 40$ optimization steps. Optimization is performed over all 16 intrinsic cloth parameters $\boldsymbol{\theta}_i$ and 1 external wind speed $\boldsymbol{\theta}_e$. We plot the estimated cloth material area weight (kg m^{-2}) and wind speed velocity (m s^{-1}), although we only have access to the true material area weight (dashed horizontal line). For this dataset, the wind speed has three settings of increasing wind speed: W1, W2 and W3. *Top row:* The cloth's true area weight is 0.17 kg m^{-2} while the final measurement attains 0.22 kg m^{-2} after only 10 iterations. *Center row:* The cloth's true area weight is 0.24 kg m^{-2} while the prediction is 0.29 kg m^{-2}. While the ground-truth wind speed is not known, the wind speed in the simulations seems like an underestimate. *Top:* A heavier cloth at 0.39 kg m^{-2} while the simulation measures 0.45 kg m^{-2}.

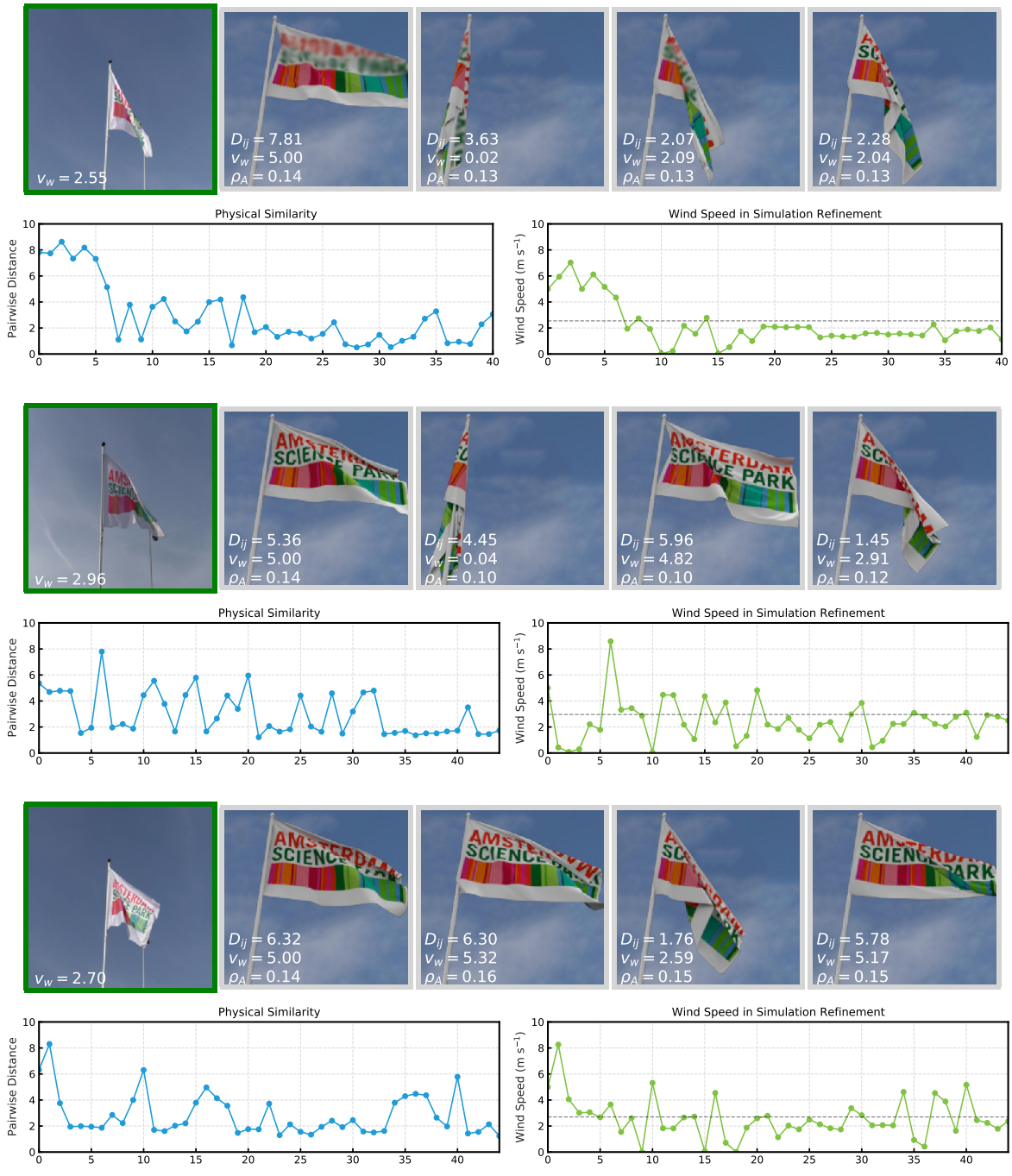

Figure 3. Additional results for our **FlagReal refined-measurements** for a random target video capturing the flag in the wind (corresponding to Figure 8 in the main paper). The five images are the center frames of the real-world target video (left) and simulations throughout the refinement process after $t = 0, 10, 20, 40$ optimization steps. Optimization is performed over all 16 intrinsic cloth parameters $\boldsymbol{\theta}_i$ and 1 external wind speed $\boldsymbol{\theta}_e$. We only visualize the simulated wind speed as it is the only parameter for which we have ground-truth (dashed line). *Top row:* successful optimization example; although the scale between real observation and simulation is different, our method is able to precisely determine the external wind speed. The real-world video has a ground-truth wind speed of 2.46 m s^{-1} while the refinement procedure finds a wind speed of 2.34 m s^{-1} in less than 10 optimization steps. *Center row:* Another successful optimization example. The real-world video has a ground-truth wind speed of 2.96 m s^{-1} while the refinement procedure finds a wind speed of 2.36 m s^{-1} after 45 refinement steps. *Bottom row:* failure case; even though after 45 steps the wind speed is approximately correct, the optimization procedure has not converged.

Listing 1. The ArcSim base **configuration for flags** [5] as JSON file to be read by the simulator. The simulation runs on a flag mesh of 3 : 2 aspect ratio in a constant wind field defined by the wind speed $\boldsymbol{\theta}_e$ parameter. During simulation we only consider a wind field in a single direction, but during rendering we use multiple relative camera orientations creating the appearance of varying wind directions. The intrinsic cloth material parameters $\boldsymbol{\theta}_i$ reside inside the material configuration file (Listing 3).

```
{
    "frame_time": 0.04,
    "frame_steps": 8,
    "duration": 20,
    "cloths": [{
        "mesh": "meshes/flag.obj",
            "transform": {
            "translate": [0, 0, 0],
            "rotate": [120, 1, 1, 1]
        },
        "materials": [{
            "data": "materials/camel-ponte-roma.json",
            "thicken": 2,
            "strain_limits": [0.95, 1.05]
        }],
        "remeshing": {
            "refine_angle": 0.3,
            "refine_compression": 0.01,
            "refine_velocity": 1,
            "size": [20e-3, 500e-3],
            "aspect_min": 0.2
        }
    }],
    "handles": [{
        "nodes": [0,3]
    }],
    "gravity": [0, 0, -9.81],
    "wind": {
        "velocity": [wind_speed, 0, 0]
    },
    "magic": {
        "repulsion_thickness": 10e-3,
        "collision_stiffness": 1e6
    }
}
```

Listing 2. The ArcSim base **configuration for hanging cloth** as JSON file to be read by the simulator. The wind speed is defined on the horizontal plane (x and y components). Again, the starting point of the intrinsic cloth material parameters $\boldsymbol{\theta}_i$ are given in Listing 3. However, in comparison to the flag simulations, we set a much larger variety of fabrics and define the fabric area weight range to correspond to the hanging cloth dataset [2].

```
{
    "frame_time": 0.04,
    "frame_steps": 8,
    "duration": 20,
    "cloths": [{
        "mesh": "meshes/square.obj",
            "transform": {
            "translate": [0, 0, 0],
            "rotate": [120, 1, 1, 1]},
        "materials": [{
            "data": "materials/camel-ponte-roma.json",
            "thicken": 1,
            "strain_limits": [0.95, 1.05]
            }],
        "remeshing": {
            "refine_angle": 0.3,
            "refine_compression": 0.01,
            "refine_velocity": 1,
            "size": [20e-3, 500e-3],
            "aspect_min": 0.2
        }
    }],
    "motions": [],
    "handles": [{"nodes": [2,3]}],
    "gravity": [0, 0, -9.8],
    "wind": {"velocity": [wind_speed_x, wind_speed_x, 0]},
    "magic": {"repulsion_thickness": 10e-3, "collision_stiffness": 1e6}
}
```

Listing 3. The **ArcSim material configuration** [5] as JSON file to be consumed by the simulator. As base material, we use "camel ponte roma" with its properties determined in the mechanical setup by [7]. This file specifies the cloth's area weight, bending stiffness coefficients and stretching coefficients. As flags are of strong, weather-resistant material, we optimize over the area weight (1×) and bending parameters (15×). Together these 16 parameters define $\boldsymbol{\theta}_i$. For hanging cloth, we also keep the bending parameters fixed to constrain the number of free parameters.

```
{
    "density": 0.135,
    "bending": [
        [36.3483e-6, 49.5855e-6, 45.7440e-6, 47.4133e-6, 20.7266e-6],
        [33.0132e-6, 29.7443e-6, 35.1036e-6, 34.0410e-6, 14.4399e-6],
        [37.1575e-6, 34.1074e-6, 33.2294e-6, 34.6855e-6, 10.4399e-6]
    ],
    "stretching": [
        [31.146198,  -12.802702,  44.028667,  31.896357],
        [78.707756,   26.754574, 268.680725,  27.743423],
        [67.368431,   77.767944, 182.273407, -14.661531],
        [113.367035,  54.802021, 175.126572,  44.657330],
        [144.294830, 111.404854, 138.422150, -29.861851],
        [143.933365,  49.654823, 191.777588,  39.491055]
    ]
}
```

Table 2. Exhaustive overview of the **render parameters** ζ for rendering the FlagSim and ClothSim datasets.

Name	**Description**	**Value/Range (Flags)**	**Value/Range (Cloth)**
`background_image`	Background image of scene	Sampled from SUN397 [9]	
`background_offset`	Background image translation	$\sim$ Uniform$(-20, +20)$	
`background_scale`	Background image scale	$\sim$ Uniform$(0.6, 1.0)$	
`sun_height`	The sun's height above the ground plane	$\sim$ Uniform$(4, 10)$	
`sun_radius`	The sun's distance to mesh	$\sim$ Uniform$(0, \ 5)$	
`sun_strength`	The sun's illumination strength	$\sim$ Uniform$(2, 10)$	
`sun_shadow_soft_size`	The sun's shadow hardness	$\sim$ Uniform$(2, 10)$	
`cycles_samples`	Cycles [1] number of render samples	50	
`cycles_bounces`	Cycles [1] light bounces, object dependent	$[0, 6]$	
`camera_height`	The height above the ground plane	$\sim$ Uniform$(0.2, 3)$	$\sim$ Uniform$(0.5, 2)$
`camera_radius`	The distance to the mesh	$\sim$ Uniform$(4, 6)$	$\sim$ Uniform$(1, 2.5)$
`camera_angle`	The orientation w.r.t. wind direction	$\sim$ Uniform$(-15, +15)$	$\sim$ Uniform$(-5, +5)$
`mesh_height`	The flag's height above the ground plane	4.6	2
`mesh_aspect_ratio`	The flag's aspect ratio	3 : 2	1 : 1
`mesh_texture`	The flag/cloth texture	Sampled from 12 countries	Sampled from [10]