

Appendix

A Dataset Generation

A.1 TOPPQuad Modifications and Dataset Generation

The LSTM models are trained on a dataset of trajectories generated from TOPPQuad [1]. On top of the dynamic feasibility constraints described in the paper (most notably bounds on the individual motor thrusts), we enforce a maximum velocity constraint v_{max} and constrain the starting yaw angle $\theta_{z,0} \in [0, \pi/2]$ to account for symmetries around the body z-axis of the quadrotor. Additionally, we add a small penalty on the total angular acceleration to the cost function to ensure consistency of optimal yaw trajectories across geometric paths that only differ by small perturbations. Therefore, trajectories in the dataset minimize the following objective function:

$$\min T + \lambda \sum_{i=0}^{N-1} \|\alpha_i\|_2^2, \quad (1)$$

where

$$T = \sum_{i=0}^{N-1} \frac{2\Delta s}{\sqrt{h_i} + \sqrt{h_{i+1}}} \quad (2)$$

and N is the number of path discretization points. We select $\lambda = 10^{-4}$. In Fig. 1, we compare the maximum difference between yaw profiles of TOPPQuad trajectories with and without this added penalty. For each datapoint, we use TOPPQuad to compute a time-optimal trajectory with and without a $[0.1, 0, 0]m$ perturbation to the first waypoint. Despite a minimal change in the h profile and total trajectory time, the maximum yaw difference clusters much more tightly around 0 with this added penalty. We posit this is due to maximum velocity being the dominant constraint in the optimization, preventing motor saturation for stretches along the path. This in turn provides the thrust authority for additional yawing without significantly impacting the total traversal time along the path. Although such an effect is innocuous when optimizing for a single trajectory, the effect is undesirable for the learning problem.

A.1.1 Simulation Experiment Dataset

For the simulation experiments, we set a maximum velocity $v_{max} = 5m/s$ for a fair comparison to the selected baselines. Waypoints are sampled from a $10m \times 10m \times 10m$ box. The models are trained on a dataset of approximately 10,000 trajectories. The trajectories for the baseline comparisons have waypoints sampled from a $10m \times 10m \times 5m$ box, again to ensure a fair comparison across baselines.

A.1.2 Hardware Experiment Dataset

For the hardware experiments, we set a maximum velocity of $v_{max} = 2m/s$, a maximum angular velocity of $\omega_{max} = 8rad/s$, and a maximum acceleration of $a_{max} = 5m/s^2$. These numbers were selected based on observation of what the CrazyFlie was physically capable of achieving, so as to minimize model mismatch. Again, waypoints for the dataset were sampled from a $10m \times 10m \times 10m$ box. Trajectories for the experiments have waypoints sampled from a $4.5m \times 8m \times 2m$ box to fit the Vicon motion capture space.

B Trajectory Recovery

Given the path of positions ($\gamma(\cdot)$), together with its first ($\gamma'(\cdot)$) and second ($\gamma''(\cdot)$) derivatives, our model outputs the square speed profile ($h(\cdot)$) and a cosine-normalized yaw ($\cos\theta_z(\cdot)$). Here, we show a sketch of how to recover the full trajectory variables from these outputs for the geometric controller [2]. We refer the reader to the [1] for full details.

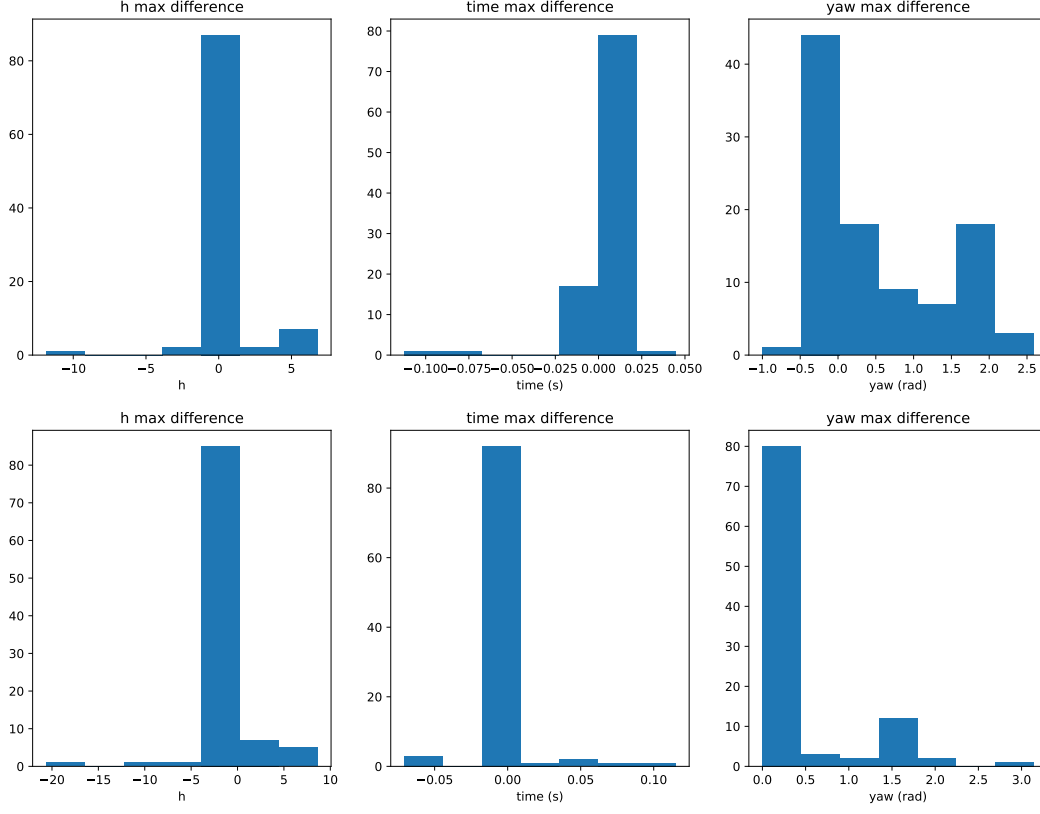


Figure 1: Maximum difference of h , time, and yaw between 100 pairs of perturbed and unperturbed TOPPQuad trajectories. **Top:** TOPPQuad trajectory comparison computed following [1]. **Bottom:** TOPPQuad trajectory comparison computed with the modifications introduced in Sec A.1. Both rows are computed with the $v_{max} = 5m/s$ constraint.

Translational Components: Given the problem formulation, we can immediately state $\mathbf{p}(t) = \gamma(\cdot)$. Differentiating this equation, we get velocity and acceleration as

$$\mathbf{v}(t) = \sqrt{h(\cdot)}\gamma'(\cdot), \quad (3)$$

$$\mathbf{a}(t) = \frac{1}{2}\gamma'(\cdot)h'(\cdot) + \gamma''(\cdot)h(\cdot). \quad (4)$$

Rotational Components: We represent the orientation of the quadrotor using the Hopf Fibration [3], where the rotation quaternion can be broken down into quaternions representing the quadrotor's body-z axis (\mathbf{b}_3) and the yaw angle.

$$\mathbf{q} = \mathbf{q}_{b_3} \otimes \mathbf{q}_{yaw} \quad (5)$$

Starting from the quadrotor's equations of motion, we know the thrust vector (\mathbf{F}) is aligned with (\mathbf{b}_3),

$$\mathbf{F} = m\mathbf{a} - \mathbf{g} \quad (6)$$

where m the mass of the quadrotor and \mathbf{g} the gravity vector. Taking the unit vector of \mathbf{F} gets us \mathbf{b}_3 , which can be computed from the previously determined acceleration (Eq. 4).

$$\mathbf{b}_3 = \frac{\mathbf{F}}{\|\mathbf{F}\|} = \frac{m\mathbf{a} - \mathbf{g}}{\|m\mathbf{a} - \mathbf{g}\|} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \quad (7)$$

Using the Hopf Fibration, we re-write this into quaternion form as

$$\mathbf{q}_{b_3} = \frac{1}{\sqrt{2(1+c)}} [1+c \quad -b \quad a \quad 0]^T \quad (8)$$

The yaw quaternion is written as

$$\mathbf{q}_{yaw} = [\cos(\theta_z/2) \quad 0 \quad 0 \quad \sin(\theta_z/2)]^T \quad (9)$$

which can be directly populated from the model output. We note that by predicting only $\cos\theta_z$, we lose information on the direction of yaw change at the $\theta_z = \pi$ boundary. However, we make the assumption that because of the added penalty on α_z in the cost function during dataset generation (Sec A.1), any change in yaw will be minimized and the yaw profile will not make sudden sharp jumps. We compute the yaw trajectory accordingly. Finally, we compute the angular velocity (ω) by inverting Eq. 23 from [1], rewritten here:

$$\mathbf{q}_{i+1} = \frac{(\mathbf{I}_4 + \frac{\Delta s}{2}\Omega(\omega_i)) \mathbf{q}_i}{\sqrt{1 + \frac{\Delta s^2}{4}\|\omega_i\|_2^2}}, \quad (10)$$

where Δs is the spacing between consecutive discretized points. By recognizing $\Omega(\omega_i)$ is the matrix multiplication equivalent to a quaternion multiplication for rotation velocity [4] (and likewise treating \mathbf{I}_4 as an identity quaternion), we can multiply \mathbf{q}_i into the numerator and set the denominator to some arbitrary variable f

$$\mathbf{q}_{i+1} = \frac{1}{f} \left(\mathbf{q}_i + \frac{\Delta s}{2} \mathbf{q}_i \otimes \begin{bmatrix} 0 \\ \omega_i \end{bmatrix} \right) = \frac{1}{f} \mathbf{q}_i \otimes \begin{bmatrix} 1 \\ \frac{\Delta s}{2} \omega_i \end{bmatrix}. \quad (11)$$

Pre-multiplying both sides by \mathbf{q}_i^{-1}

$$\mathbf{q}_i^{-1} \otimes \mathbf{q}_{i+1} = \frac{1}{f} \mathbf{q}_i^{-1} \otimes \mathbf{q}_i \otimes \begin{bmatrix} 1 \\ \frac{\Delta s}{2} \omega_i \end{bmatrix} = \frac{1}{f} \begin{bmatrix} 1 \\ \frac{\Delta s}{2} \omega_i \end{bmatrix}. \quad (12)$$

Let $\mathbf{q}_i^{-1} \otimes \mathbf{q}_{i+1} = \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix}$. We use the scalar component of this quaternion to solve for f then isolate ω_i from Eq. 12 to get

$$\omega_i(\cdot) = \frac{2\mathbf{q}_v}{\Delta s q_w}. \quad (13)$$

Re-scaling $\omega_i(\cdot)$ with h to fit the learned time profile as a parallel to Eq. 3, we finally get

$$\omega_i(t) = \sqrt{h} \frac{2\mathbf{q}_v}{\Delta s q_w}. \quad (14)$$

C Additional Experiments

C.1 Input Ablation Study

The goal of this experiment is to compare the performance of the LSTM architecture given different sets of inputs. Using the same metrics as prior ablation experiments, we compare the performance of LSTM models trained on combinations of the position, initial velocity, and initial acceleration of dataset trajectories in Table 1. By all four metrics, its clear that position alone is insufficient. The combinations of position, velocity, and acceleration all perform similarly, but we select the **pos/vel/acc** model due to its lower TD Ratio and failure rate with minimal losses in simulation performance.

	TOPPQuad	pos/vel/acc		pos/vel		pos/acc		pos	
		Train	Test	Train	Test	Train	Test	Train	Test
max dev (m)	0.053	0.074	0.143	0.106	0.148	0.068	0.164	0.097	0.225
thrust violation (N)	0.000	0.002	0.009	0.002	0.005	0.003	0.007	0.008	0.026
TD ratio (%)	5.929(s)	-0.70%	-0.40%	-0.70%	-1.9%	-0.90%	-2.10%	-0.70%	-1.90%
failure (%)	0.0%	2.0%	4.0%	4.0%	4.0%	2.0%	6.0%	8.0%	22.0%

Table 1: **Ablation study on model architectures.** Each statistic is averaged over 100 trajectories. A negative *TD ratio* ($\frac{\text{PRED. TIME} - \text{OPT. TIME}}{\text{OPT. TIME}}$) denotes a shorter travel time relative to TOPPQuad (OPT. TIME).

C.2 Baseline Paths

In this section, we select successful trajectories from all three baselines to illustrate the change in path shape and length (Fig. 2). Our planner (and TOPPPQuad) determines a dynamically-informed path using the minimum snap trajectory planner [5], given a set of waypoints and a set of time segments calculated by the euclidean distance between consecutive waypoints and a nominal velocity ($v_{nom} = 1m/s$). MFBO [6] likewise uses a minimum snap planner with the same waypoints, but attempts to find a better set of time segments that is time optimal while respecting quadrotor dynamics. Finally, AllocNet [7] keeps the start and end waypoints for a minimum jerk trajectory, but attempts to predict both intermediary waypoints and the time allocation for a time-optimal trajectory from a set of convex corridors.

For the AllocNet planner, we fit three bounding boxes to evenly spaced sections of the original minimum snap path, oriented to the plane of each path sections and with a $0.75m$ padding from any box wall to the path. The exterior of these bounding boxes are then converted into a points cloud, from which AllocNet generates a set of convex corridors. We select the padding to ensure the convex corridors generated are generous enough to not lie outside of the planner’s training set for a fair comparison.

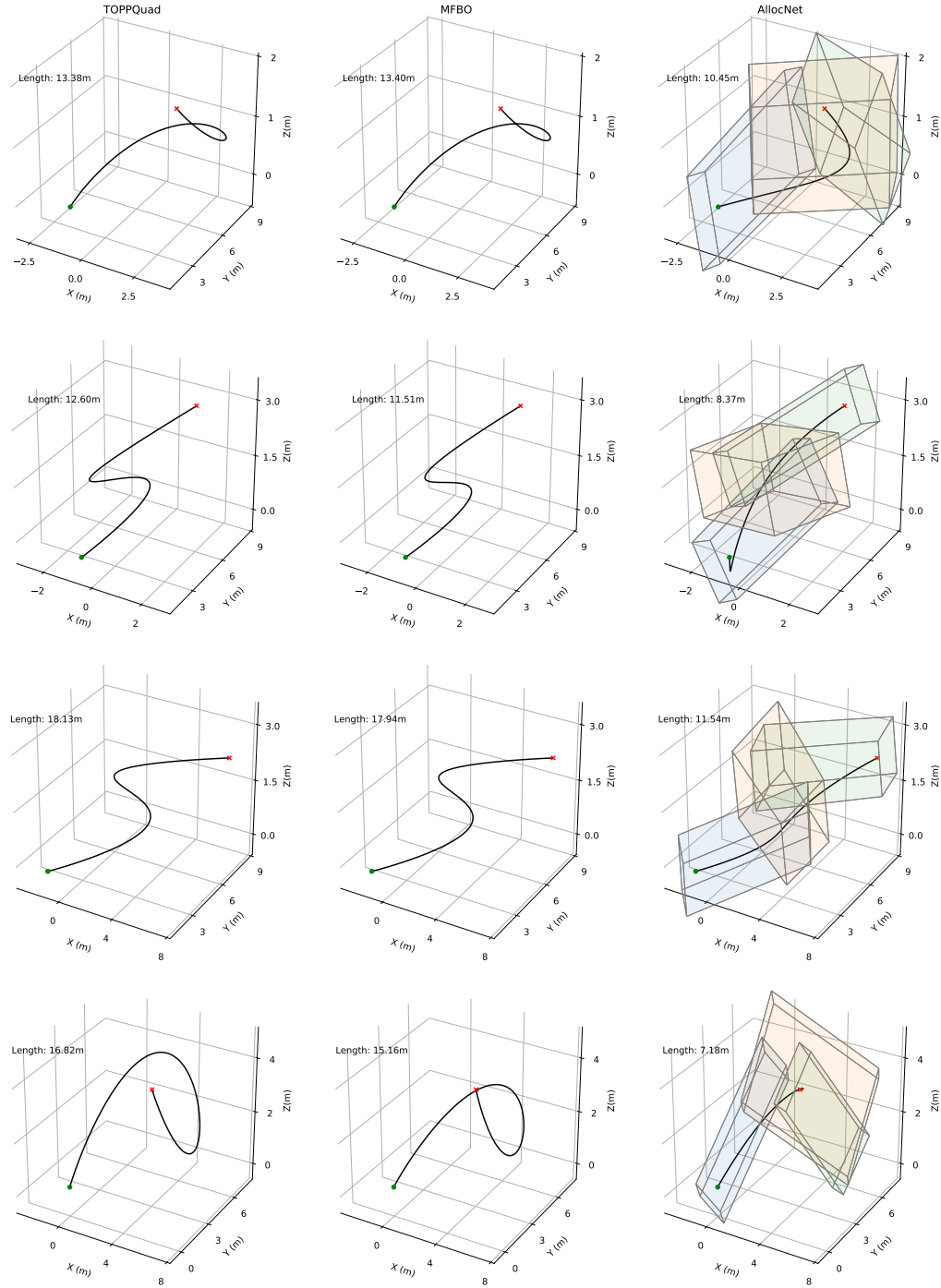


Figure 2: Sample paths generated by our planner (**left**), MFBO (**middle**) and AllocNet (**right**). We also show the bounding boxes given to the AllocNet planner.

C.3 Noised Model Hardware Experiments

In this section, we present additional hardware experiments with trajectories predicted from the LSTM-0.1 (Fig. 3) and LSTM-0.01 models (Fig. 4). Each subfigure represents 5 trials of the learned trajectory and 4 trials of the TOPPQuad trajectory. Table 2 compares the performance of these additional models with the LSTM model presented in the main paper. We show all three models have good tracking performance and similar travel times. However, we note that for the **LSTM-0.01** model, all trials of 'Traj 1' crashed. While the trajectory itself appears very simple, a model trained on a dataset of polynomial paths would be more likely to struggle generalizing to a straight line path.

	TOPPQuad	LSTM	LSTM-0.1	LSTM-0.01
max deviation (m)	0.347	0.355	0.373	0.372
travel time (s)	7.981	8.355	8.288	8.114

Table 2: Hardware experiments.

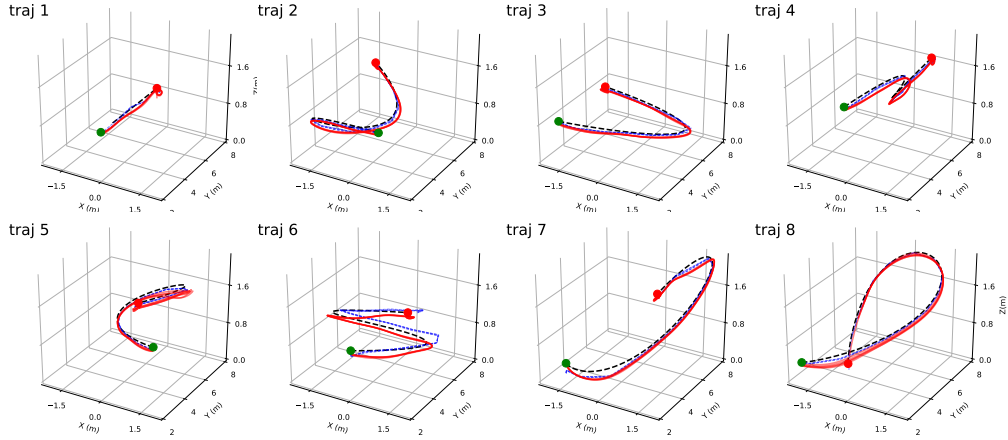


Figure 3: Experiment visualization of trajectories predicted from the **LSTM-0.1** model, plotted from the collected motion capture flight data. The dashed black line represents the reference geometric path, the dashed blue line shows the tracked trajectory generated by TOPPQuad, and the solid red lines show the tracked trajectories output by our model across different runs.

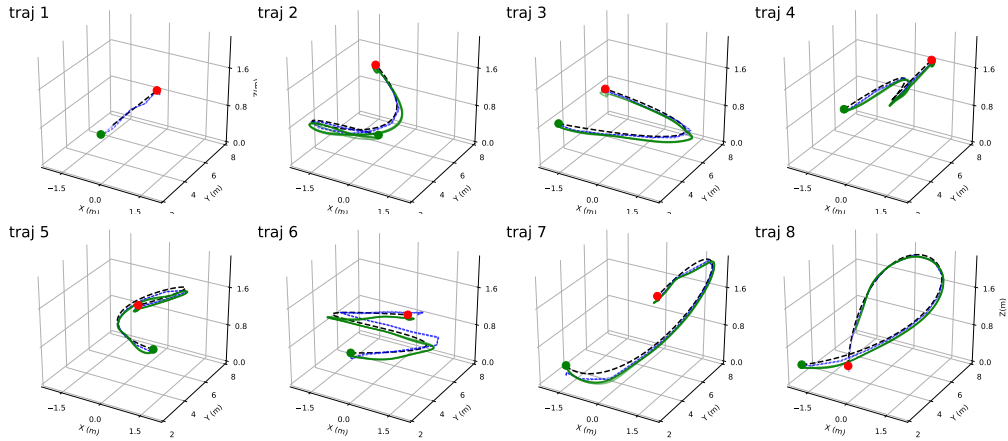


Figure 4: Experiment visualization of trajectories predicted from the **LSTM-0.01** model, plotted from the collected motion capture flight data. The dashed black line represents the reference geometric path, the dashed blue line shows the tracked trajectory generated by TOPPQuad, and the solid green lines show the tracked trajectories output by our model across different runs. All trials of Traj 1 from the predicted model crashed.

References

- [1] K. Mao, I. Spasojevic, M. A. Hsieh, and V. Kumar. Toppquad: Dynamically-feasible time-optimal path parametrization for quadrotors. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13136–13143, 2024. doi:[10.1109/IROS58592.2024.10801611](https://doi.org/10.1109/IROS58592.2024.10801611).
- [2] T. Lee, M. Leok, and N. H. McClamroch. Geometric tracking control of a quadrotor uav on $se(3)$. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5420–5425, 2010. doi:[10.1109/CDC.2010.5717652](https://doi.org/10.1109/CDC.2010.5717652).
- [3] M. Watterson and V. Kumar. Control of quadrotors using the hopf fibration on $so(3)$. In *Robotics Research: The 18th International Symposium ISRR*, pages 199–215. Springer, 2019.
- [4] B. Graf. Quaternions and dynamics, 2008. URL <https://arxiv.org/abs/0811.2889>.
- [5] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE International Conference on Robotics and Automation*, pages 2520–2525, 2011. doi:[10.1109/ICRA.2011.5980409](https://doi.org/10.1109/ICRA.2011.5980409).
- [6] G. Ryou, E. Tal, and S. Karaman. Multi-fidelity black-box optimization for time-optimal quadrotor maneuvers. *The International Journal of Robotics Research*, 40(12-14):1352–1369, 2021.
- [7] Y. Wu, X. Sun, I. Spasojevic, and V. Kumar. Deep learning for optimization of trajectories for quadrotors. *IEEE Robotics and Automation Letters*, 9(3):2479–2486, 2024.