Supplemental document accompanying submission to *Optics Express*

**Title:** Inferring the solution space of microscope objective lenses using deep learning

**Authors:** Geoffroi Côté, Yueqian Zhang, Christoph Menke, Jean-François Lalonde, Simon Thibault

**Submitted:** 12/19/2021 9:42:59 PM
Inferring the solution space of microscope objective lenses using deep learning: supplemental document

This document provides additional methodology details and results related to the article "Inferring the solution space of microscope objective lenses using deep learning."

1. MICROSCOPE OBJECTIVE LENS SPECIFICATIONS

In Table S1 are shown the complete specifications for the nominal design case. Those specifications were used as a basis for our unified MOL framework, in particular in the choice of the OLF and training domain.

| Parameter          | Original              | Converted                                                                 |
|--------------------|-----------------------|---------------------------------------------------------------------------|
| Magnification      | 20x                   | Normalized focal length EFL = 1                                          |
| Conjugate          | Infinite              | Reversed design, infinite conjugate                                       |
| Tube lens          | 200 mm, solo correction | No tube lens, solo correction                                               |
| Field size         | 25 mm intermediate image size | ±3.57° field angle                                                          |
| NA                 | 0.4                   |                                                                           |
| Spectrum           | C, d, F, g (Apo)      |                                                                           |
| Color correction   | Apo                   | Full field, full spectrum                                                 |
| Field correction   | Plan                  | RMS spot size as small as possible                                         |
| Distortion         | < 1 %                 |                                                                           |
| Cover glass        | None                  |                                                                           |
| Parfocal length    | 60 mm                 | TTL < 6                                                                   |
| Working distance   | 3 mm, 10 mm (LD)      | 0.3, 1 (LD)                                                               |
| Glass catalog      | Schott + Ohara        |                                                                           |
| Lens geometry      |                       |                                                                           |
| – Central thickness| < 1/7 diameter        |                                                                           |
| – Edge thickness   | < 0                   |                                                                           |

Table S1. List of specifications for the nominal case. When necessary, we convert the original specifications to our unified framework for MOLs.

2. COMPLETE OPTICAL LOSS FUNCTION FORMULATION

Here, we detail the optical loss function (OLF) used in this work. The core component of the OLF is the weighted spot size radius \( r \), which is sampled at \( \{0, 0.7, 1\} \) times the HFOV with weights \( \{2, 1, 1\} \). As in [1], we add penalty terms that target various constraints. The ray behavior penalty \( p_{\text{behavior}} \) is used to avoid ray failures and overlapping surfaces. The glass penalty \( p_{\text{glass}} \) targets the distance between each inferred set of intermediate glass variables and the closest catalog glass from the Schott and Ohara catalogs. The shape penalty \( p_{\text{shape}} \) targets the diameter-to-thickness ratio of the glass elements and constrains it between a lower threshold, set to 3, and an upper threshold, set to 7. We refer readers to [1] for more details on the ray-tracing and ray-aiming process, as well as the computation of the RMS spot size radius and three penalty terms.

To adapt our OLF to MOLs and fulfill the aforementioned specifications, we add new penalty terms that target the working distance (WD) and total track length (TTL).
The working distance penalty $p_{WD}$ encourages the model to increase the WD of the output designs. We design this penalty term so that its slope diminishes monotonically with the WD, which allows the model to strike a balance between optical performance and WD. We compute the penalty as follows:

$$p_{WD} = \exp\left(-\frac{WD}{WD_0}\right), \quad (S1)$$

where $WD_0$ controls the shape of the penalty, which we set at 0.5.

The total track length penalty $p_{TTL}$ is used to provide an upper threshold $TTL_{max}$, using the ramp function $R$:

$$p_{TTL} = R(TTL - TTL_{max}). \quad (S2)$$

We set $TTL_{max} = 6$, which corresponds to a maximum total track length of 60 mm in the nominal case with EFL = 10 mm.

To combine the weighted spot size radius $r$ and penalty terms into our OLF, as in [1], we use multiplicative factors instead of additive factors, which makes it easier to scale different penalty terms across multiple designs. The OLF for a given design is computed as follows:

$$OLF = r \prod_i (1 + \lambda_i p_i), \quad (S3)$$

where each penalty term $p_i$ is scaled by a factor. We use $\lambda_{behavior} = 10^4$, $\lambda_{glass} = 32$, $\lambda_{shape} = 10$, $\lambda_{WD} = 6$, and $\lambda_{TTL} = 1$.

3. DETAILED MODEL ARCHITECTURE

The dynamic model architecture used in this work (see Fig. S1) is similar to the one in [1]. It is also functionally equivalent except for the additional output branches.

Dynamic network. As in [1], we use an off-the-shelf dynamic network that can operate on sequences with different lengths. However, we replaced the recurrent neural network (RNN) with a transformer encoder [2], with 6 layers, a dimensionality $d_{model} = 256$ in the inputs and 512 in the fully-connected layers, 8 attention heads, and no dropout. We use the “Pre-LN” configuration for the normalization layers [3]. The dynamic network expects inputs of dimensionality $L \times d_{model}$, where $L$ is the length of the sequence, and returns outputs of the same shape. We use learned embeddings and a projection layer to transform our inputs into the required format.

Learned embeddings. As in [1], we use the extended sequence, given as input, to dynamically add, remove, or rearrange the components of the model—this mechanism allows the model to adapt its predictions depending on the desired lens sequence. Each element of the extended sequence is one of four types of basic structure: an interface “i”, a glass element “g”, an air gap “a”, or the aperture stop “s”. The extended input sequence achieves the same results as in [1] while being more compact. Each basic structure is associated with an embedding vector $e$ of size $d_{model}$ that is learned along with the other model components.

Positional encoding. Unlike RNNs, transformer models have no inherent mechanism to infer the order of the sequence of inputs, e.g. to distinguish the sequence “i-g-s-i-a” from “s-i-g-i-a”. Similar to most transformer approaches, we add to the learned embeddings a “positional encoding” of size $d_{model}$ that encodes the index of each element of the sequence. In contrast to common practice, the positional encoding depends not only on the index in forward order, but also in backward order (from the last to the first optical surface), which can help the model locate the last elements of the lens design more easily.

Specifications. The specifications (EPD, HFOV) form a 2-dimensional vector that is linearly projected into a vector of size $d_{model}$ using the learned linear layer $m_p$, and passed to the dynamic network as an additional element in the sequence of inputs. This formulation is more natural for transformer architectures than the one used with the previous RNN model [1]: passing the specifications through a linear layer for each type of basic structure, then feeding those embeddings to the dynamic network.

Output layers. The output layers linearly project the outputs of the dynamic network of size $d_{model}$ into the appropriate dimensionality for each type of output: curvatures, intermediate glass variables, and intermediate thicknesses. There are no outputs for the aperture stop and...
specification vector
extended sequence
learned embeddings
positional encoding
dynamic network
output layers
intermediate lens variables

EPD, HFOV

\[ i \quad g \quad s \quad i \quad a \]

\[ e_i \quad e_g \quad e_s \quad e_i \quad e_a \quad m_p \]

\[ e_{1,5} \quad e_{2,4} \quad e_{3,3} \quad e_{4,2} \quad e_{5,1} \]

\[ m_d \]

\[ m_{o,1} \quad m_{o,g} \quad m_{o,a} \]

\[ \epsilon_1 \quad \epsilon_1 \quad \epsilon_1 \quad \epsilon_1 \quad \epsilon_1 \]

\[ n_C, n_{g1}, n_F, n_g \]

\[ t_1 \quad t_2 \quad t_1' \quad t_2' \]

**Fig. S1.** Dynamic model architecture, illustrated for a single-lens design (adapted from [1]).

a) The inputs include the specification vector as well as the extended sequence composed of four types of basic structures: interface “i”, glass element “g”, air gap “a”, or aperture stop “s”. The extended sequence is used to dynamically rearrange the learned embeddings and output layers. b) The backbone of the model is the dynamic network. The learned embeddings \( e \) and projection layer \( m_p \) transform the inputs into an intermediate, high-dimensional representation that can be processed by the dynamic network. The output layers linearly project the high-dimensional outputs of the dynamic network into intermediate lens variables. Each box represents a learnable component of the model. c) The intermediate lens variables are transformed into meaningful lens variables using non-learnable operations. In this representation, there is only one output branch \((K = 1)\); otherwise, the output layers would have \( K \) times as many outputs.

specifications: they are only fed as inputs so that the model can adapt its predictions. Likewise, there are no outputs for the last curvature, since it is computed directly to enforce a unit EFL. The number of outputs of these layers is proportional to the number of output lens branches \( K \) (not shown in Fig. S1).

4. TRAINING DETAILS

In unsupervised training, we empirically found it helpful to sample the lens sequences of the reference designs more often, so that the model quickly learns to extrapolate from those reference designs. Therefore, every time we generate an unsupervised batch, we sample the lens sequences that are common to both unsupervised and supervised training 100x more often than the others. To train the model, we use the Adam optimizer with parameters \( \beta = (0.9, 0.99) \) and use gradient clipping with an upper threshold of 0.1. We train over 500 000 training steps. The learning rate is linearly increased from \( 10^{-6} \) to \( 10^{-3} \) during the first 10 000 steps, then progressively decayed to 0 over the rest of the training steps using a cosine half-cycle.

5. ABLATION STUDY

In Table S2, we present an ablation study to justify some of our methodology choices. For each experiment, we train the model using the exact procedure explained herein except that we exclude one aspect of our method. Here, we consider methodology choices that aim to help the training process increase the overall optical performance of the designs: the paraxial image solve (PIM)—used to output the defocus rather than the thickness between the lens and
image plane—and the supervised loss. We report the geometric average of the OLF across all
lens sequences and branches for the nominal specifications, for a total of 59,456 designs. Our
results show that excluding the PIM leads to significantly worse designs as the average OLF goes
up by 1.6x. Despite the structural losses providing a supervision signal on the lens structures,
supervised training is still mandatory as its exclusion brings the average OLF up by 6.8x.

| Methodology choice                        | OLF          | OLF (relative) |
|-------------------------------------------|--------------|----------------|
| Baseline                                  | $1.61 \times 10^{-3}$ | 1x             |
| Without paraxial image solve (PIM)        | $2.64 \times 10^{-3}$ | 1.6x           |
| Without supervised training ($\lambda_S = 0$) | $1.09 \times 10^{-2}$ | 6.8x           |

Table S2. Ablation study. We report the geometric average of the OLF across all branches and
lens sequences for the nominal specifications.

6. ADDITIONAL RESULTS

In the following, as previously, we query the model using the nominal specifications ($NA = 0.40$,
$HFOV = 3.57 \text{ deg}$) and scale the designs to $EFL = 10 \text{ mm}$.

In Fig. S2 are shown additional layouts of designs inferred using the nominal specifications.
Here, instead of selecting the lens sequence that minimizes the OLF for a given branch and
number of glass elements, we randomly select a lens sequence out of all candidates with the
proper amount of glass elements.

Fig. S2. Subset of random designs inferred from all branches $k$ of the trained model, using
the nominal specifications (the scale is in units of EFL). For each branch, only 6 sequences are
shown out of 7432 candidates.

In Fig. S3 is shown the distribution of the distortion for all designs. Most designs have standard
distortion values within $\pm 1 \%$ regardless of the branch and number of elements.

In Fig. S4, we show the distribution of the glass variables inferred by the model across all lens
sequences and output branches, for the nominal specifications. We compare those to the catalog
Fig. S3. Distribution of the distortion of all inferred designs, grouped by number of elements and output branch, for the nominal case.

glasses that were used to formulate the glass penalty in the OLF as well as fit the glass model. For the vast majority of inferred designs, the glass variables lie close to the selected catalog glass materials used in our OLF. Note that we display the Abbe number and partial dispersion of the glass materials even though those traditional glass variables are bypassed in our framework, since we output the refractive indices at the desired wavelengths directly. When replacing the inferred glass variables with their closest glass catalog counterparts, we observe only a modest decrease in performance as the weighted RMS spot size radius $r$, geometrically averaged across all designs, goes up by 4.2%.

Fig. S4. Glass materials of the designs inferred from all 7432 lens sequences and 8 branches. For reference, we include the catalog glasses that were used during training.

For the nominal case, 95.2% of the inferred designs completely fulfill the NA and HFOV, meaning that all rays were traced successfully without encountering ray failures (either missed surfaces or total internal reflection) or traveling backwards (overlapping surfaces).

REFERENCES
1. G. Côté, J.-F. Lalonde, and S. Thibault, “Deep learning-enabled framework for automatic lens design starting point generation,” Opt. Express 29, 3841–3854 (2021).
2. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in neural information processing systems, vol. 30 (2017), pp. 5998–6008.
3. R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, “On layer normalization in the transformer architecture,” in International Conference on Machine Learning, (PMLR, 2020), pp. 10524–10533.