Spatio-Temporal Turbulence Mitigation: A Translational Perspective

Supplementary Material

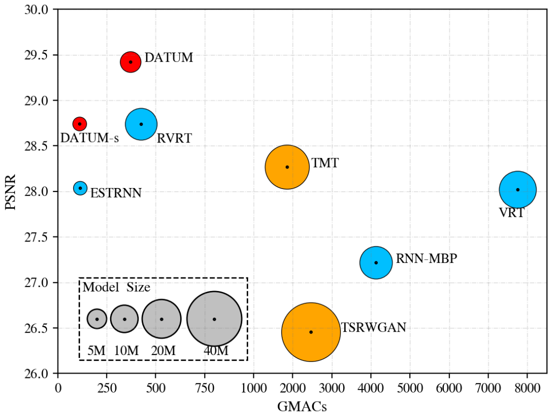


Figure 1. Benchmarking video restoration models for turbulence mitigation on our ATSyn-dynamic dataset. The circles in orange are other video-based TM networks, and the circles in blue are representative video deblurring and general restoration networks. GMACs are evaluated on 540 × 960 images.

1. Additional Experiments

1.1. DATUM-s

To further substantiate the efficacy of DATUM's design, we introduced a scaled-down variant, DATUM-s. The performance of DATUM-s is demonstrated in Fig. 1 and Table 1. Although DATUM-s retains the fundamental architecture of DATUM, it operates with only half the number of channels. This reduction assesses the model's performance under constrained computational resources, offering insights into its scalability and efficiency.

1.2. Visualization of flow refinement in DAAB

The Deformable Attention Alignment Block (DAAB) is designed to align features from a current time frame, denoted as time t, with reference features from a preceding frame, time $t-1$, during forward temporal propagation. This approach differs fundamentally from traditional optical flow methods, which align two degraded frames between times t and $t-1$ by $O^f_{t\to t-1}$. DAAB instead aligns the feature map of the current frame t with a potentially tilt-corrected reference feature from the previous frame $t-1$. The effectiveness of DAAB has been substantiated in previous ablation studies.

To further illustrate its efficacy, we provide an additional visualization in Fig. 2, leading to several critical observations:

1. The original flow estimation $O^f_{t\to t-1}$ captures mild motion, such as that of a person, but introduces noise due to random pixel displacements in static image regions.
2. The refined flow that registers f_t to r_{t-1} is more dependent on the structural information and less sensitive to the mild motion.
3. The magnitude of the refined flow under DAAB exhibits a pattern indicative of tilt rectification.
4. Additional visualization of the estimated reverse tilt field $\hat{\ }_t^{-1}$, which adjusts frame t to a tilt-free state, demonstrates that $O_t^{f\to r}$ aligns more closely with $\hat{\ }_t^{-1}$. This alignment is in line with the intended design of DAAB for effective feature-reference registration.

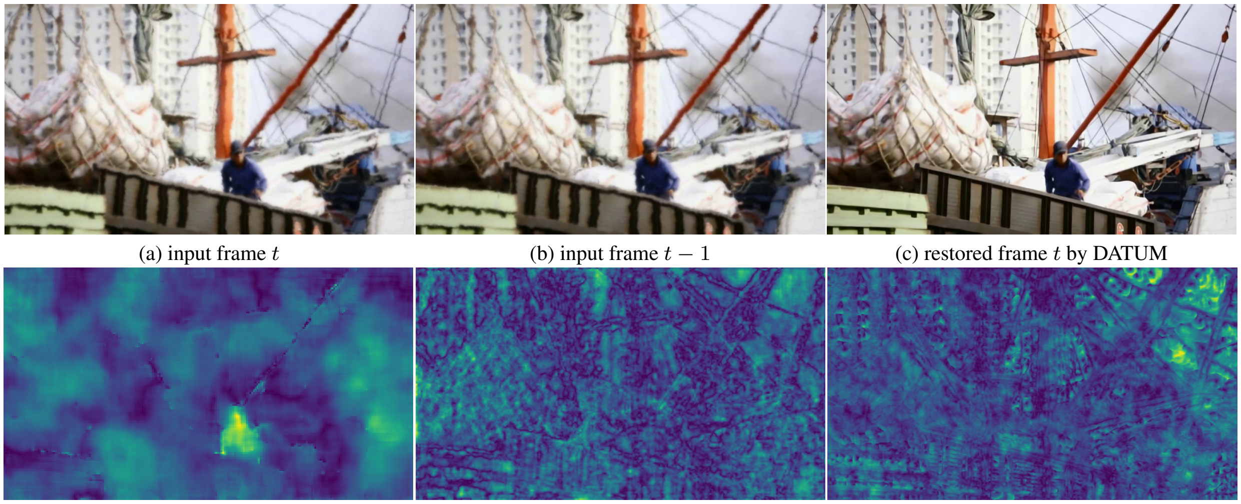
1.3. More qualitative comparisons on real-world image sequences

ATNet [10] on the static scene data. In Fig. 6, we show the restoration results of NDIR [8] rather than the ATNet [10]. NDIR is an unsupervised multi-frame pixel alignment network without a deblurring function, while ATNet is a single-frame-based general TM network. However, ATNet's inference is not successful. The results on some static scene data are shown in Fig. 3, which suggests it is challenging for this single-frame-based model to deal with medium to strong turbulence, while our methods can handle much wider turbulence conditions.

Compare with TSRWGAN [7] We address the generalization facilitated by our data synthesis method. A qualitative comparison was made between the original TSRWGAN [7] and our fine-tuned version on [7]'s real-world dynamic scenes along with a cross-dataset evaluation between these two versions on [1]'s real-world dynamic scenes. The result is shown in Figure 4. The original model shows a limit in generalization when adapting to a different dataset, but our fine-tuned version is more generalizable due to ATsyn's wide range of turbulent conditions. The original TSRWGAN model is trained from the simulator from [12] and physical simulation by heating the air along a relatively short path. Their numerical simulator can generate physics-based tilt and spatially varying blur, but higher-order aberrations are not modeled. Their physical simulator tends to generate spatially highly correlated distortion but a weak blurry effect. Because of these limitations in their generation, their generalization to other datasets suffers as a result. **Compare with Complex-CNN [1]** A complex-valued convolutional neural network (CNN) [1] was proposed to remove turbulence-related degradation from videos. Their synthetic training data comes from a simulator that models the tilt and blur via a low-order approximation, with the

Turbulence Level	Weak		Medium		Strong		Overall		Cost	
Methods	PSNR	SSIM$_{\text{CW}}$	PSNR	SSIM$_{\text{CW}}$	PSNR	SSIM$_{\text{CW}}$	PSNR	SSIM$_{\text{CW}}$	Size	FPS
DATUM-s [ours]	29.5958	0.8809	28.9869	0.8762	27.5456	0.8550	28.7743	0.8714	2.538	22.48
DATUM [ours]	30.2058	0.8857	29.6203	0.8829	28.2550	0.8640	29.4222	0.8781	5.754	9.17

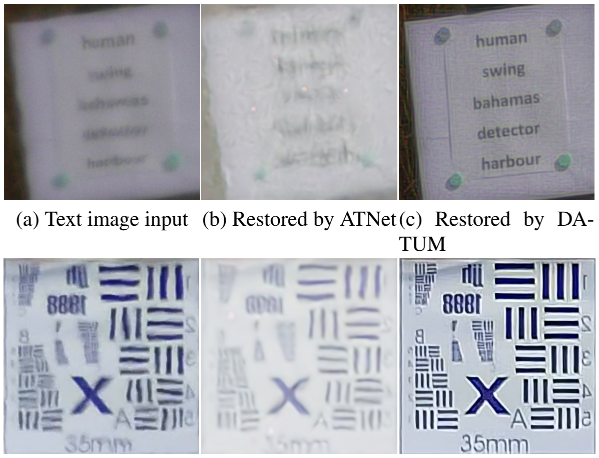
Table 1. Performance comparison on the ATSyn-dynamic set, we list the image quality scores on different turbulence levels and frame-wise resource consumption (measured with 960×540 frame sequences on RTX 2080 Ti).



(a) input frame t (b) input frame $t-1$ (c) restored frame t by DATUM

(d) optical flow from t to $t-1$ $O^{f}_{t\to t-1}$ (e) refined feature to reference flow $O^{f\to r}_{t}$ (f) estimated inverse tilt field$\hat{}^{-1}_{t}$

Figure 2. Visualization of the flow refinement for feature-reference registration in DAAB. (d), (e) and (f) show the magnitude of the associated deformation field. We ignore the directional information because it is relatively random. Note both (d) and (e) are measured in $1/4$ resolution, while (f) is in full resolution, which aims to register shallow features extracted from (a) those from (c).



(a) Text image input (b) Restored by ATNet (c) Restored by DATUM

(d) OTIS input (e) Restored by ATNet (f) Restored by DATUM

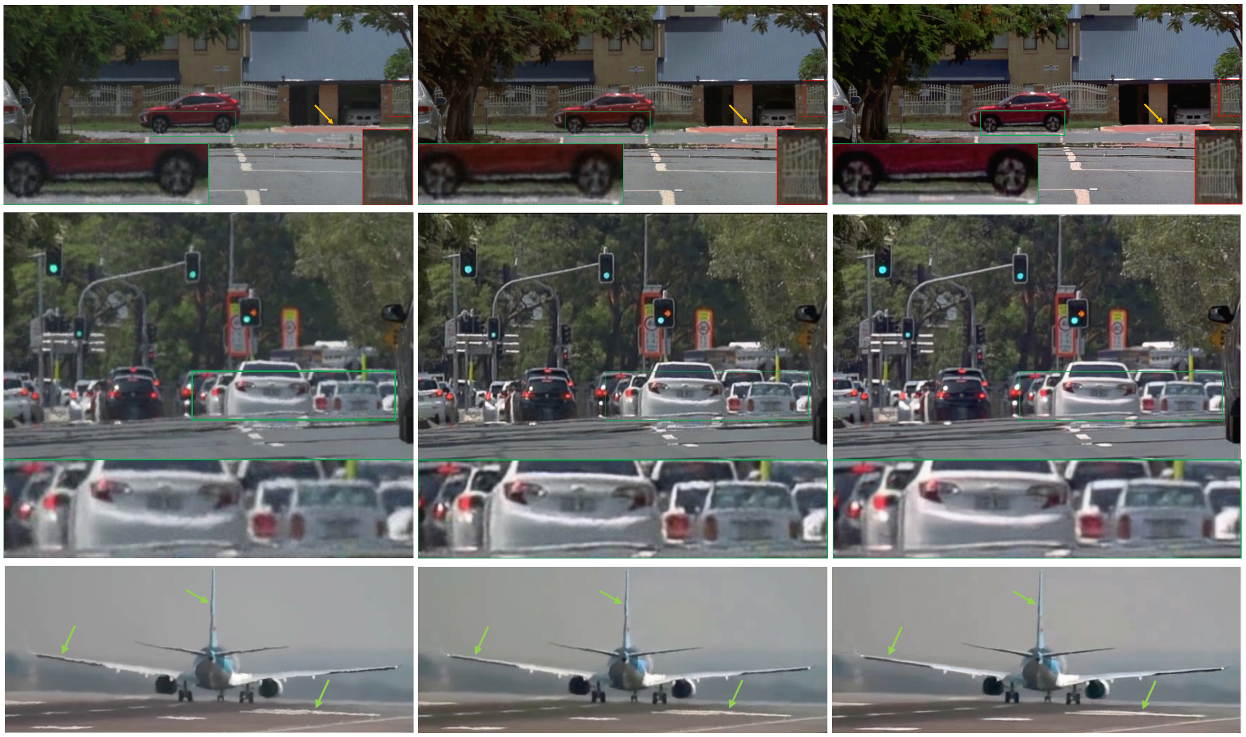
Figure 3. Cases of ATNet [10] restoration on real-world static scene images. The text image is the 49th frame of the 94th sequence in [14], and the OTIS image is the 24th frame of the pattern 13 from the [6] dataset.

blur kernel being sampled from 9 given point spread functions. Without access to their trained model, we cannot fine-tune. However, with some results available, we may compare the performance of our restored videos with theirs on their dataset. We provide this comparison in Figure 5.

1.4. Image quality metrics for turbulence mitigation

In our empirical study, we observed a high correlation between two commonly used metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). Atmospheric turbulence typically induces blur and pixel displacement in images. While the blurring effect is readily noticeable in both human and computer vision applications, minor pixel displacements often remain less perceptible. However, PSNR and SSIM are particularly sensitive to pixel displacements. This sensitivity raises the need for additional metrics to enable a more comprehensive performance evaluation. We investigate the Complex-wavelet SSIM (CW-SSIM), a variant of SSIM that is less sensitive to mild pixel displacement, and the Learned Perceptual Image Patch Similarity (LPIPS) for this purpose.

With the turbulence simulator detailed in the section 2, we can synthesize different levels of atmospheric turbulence. For the Zernike-based simulator, the turbulence effect can be quantified by the magnitude of Zernike coefficients, which indicate the properties of phase distortion

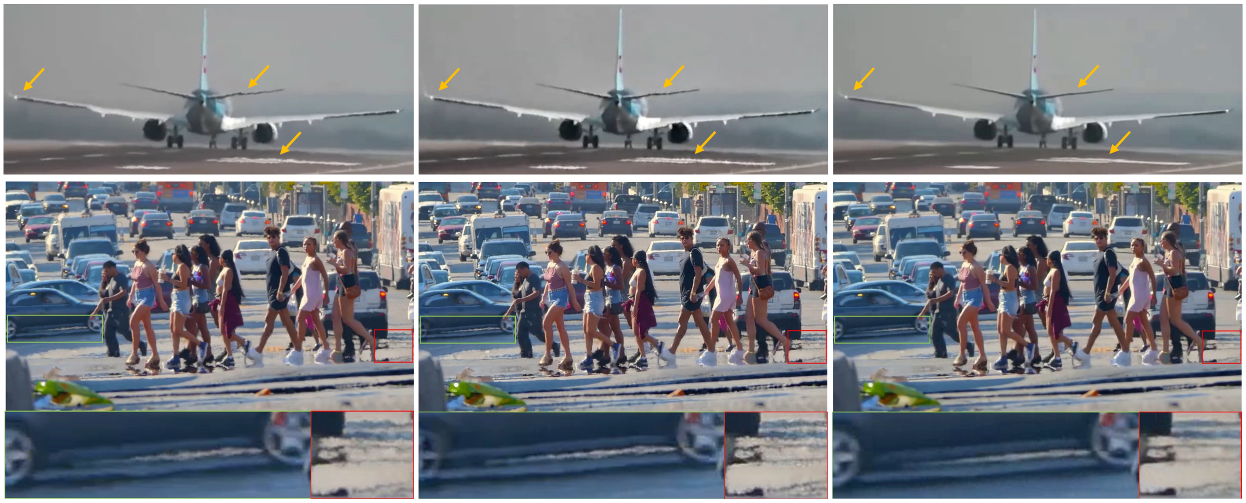


(a) input

(b) Original TSRWGAN [7]

(c) Fine-tuned TSRWGAN

Figure 4. Compare the TSRWGAN [7] trained on the original dataset and our dataset, the first two rows are real-world samples from [7]'s dataset, and the bottom row is from [1]'s real-world videos. In column (c), we present the fine-tuned TSRWGAN on our ATSyn-dynamic dataset. From the comparison, it's easy to conclude that our ATSyn dataset helps the previous turbulence mitigation network generalize better on their own testing videos and other samples.



(a) input

(b) Restored by Complex-CNN [1]

(c) Restored by DATUM

Figure 5. Comparison with [1] on their real-world dataset, zoom in for a better view.

caused by anisoplanatic turbulence. We compute different image quality scores for each pair of degraded and clean images. To assure the robustness of our analysis, we randomly chose 1000 images from the Places dataset [17] as clean images and simulated nine degraded samples for each, so we draw 9000 samples in total and show the relationship between the strength of turbulence degradation and image quality metrics in Fig. 6. Note we separate the tilt and blur effects, although they are highly correlated. The score of tilt is the average magnitude of pixel displacement on an

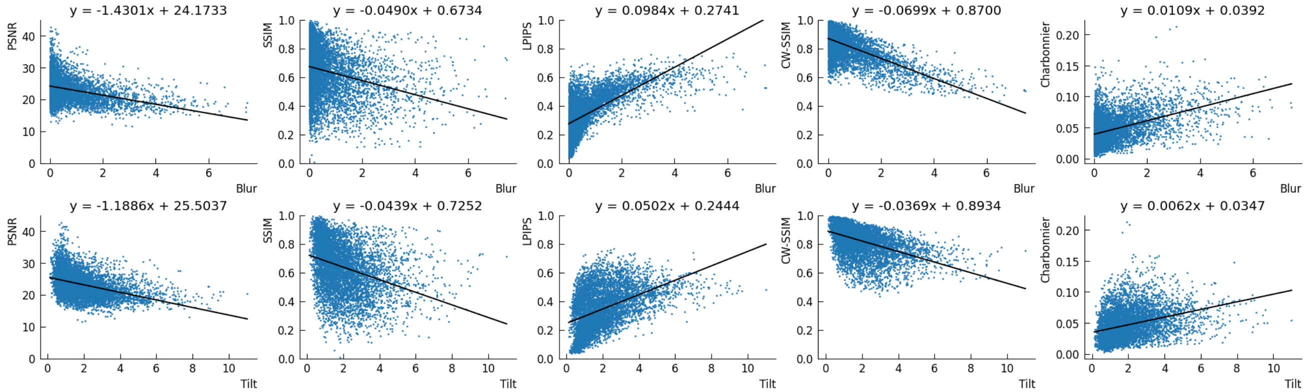


Figure 6. Image quality metrics. The x-axis is the score of blur or tilt; y-axis is the image quality score measuring the degradation with respect to the clean image. We measured PSNR, SSIM, LPIPS, CW-SSIM, and the Charbonnier score, which serves as the loss of our optimization for turbulence mitigation.

image, and the score of blur is calculated by

$$\text{blur} = k_b \frac{\sum_{\mathbf{x}}(\sqrt{\sum_{i=3:36} a_{\mathbf{x},i}^2})}{HW},$$

where $\mathbf{x} = (x, y)$ is the pixel coordinate on each image, H, W are the height and width of the image, and k_b is the scaling factor determined by the relative size of blur kernels.

From Fig. 6, we can find the SSIM is less sensitive to turbulence degradation than the others, and CW-SSIM is more sensitive than LPIPS. Thus, we selected PSNR and CW-SSIM as our restoration quality estimators.

2. Zernike-based Turbulence Simulator

2.1. General Theory

We adopt the model of the atmospheric degradations to be exclusively phase distortions, which can be represented via the Zernike polynomials $\{\mathbf{Z}_i\}$ as a basis, with coefficients $\mathbf{a}_{\mathbf{x},i}$ [4, 11]. We set $i \in \{1, 2, 3, \cdots, 36\}$ with $\mathbf{Z}_{\{2,3\}}$ influencing the pixel displacement $\boldsymbol{\mathcal{T}}$ and higher order coefficients $\mathbf{Z}_{\{i\geq 4\}}$ forming the blurry effect $\boldsymbol{\mathcal{B}}$ in the image plane. With this, the kernel of $\boldsymbol{\mathcal{B}}_{\mathbf{x}}$ can be written as:

$$\boldsymbol{\mathcal{B}}_{\mathbf{x}} \approx \left| \mathcal{F}\left\{ \exp\left(-j \sum_{i=4}^{36} a_{\mathbf{x},i}\mathbf{Z}_i \right) \right\} \right|^2, \quad (1)$$

where $\mathcal{F}$ denotes the Fourier transform. Adopting the wide sense stationary model for the Zernike coefficients [4, 5], one can generate $\mathbf{a}_{\mathbf{x},i}$ in parallel by Fourier Transform. It is worth noting $\mathbf{a}_{\mathbf{x},\{2,3\}}$ can be excluded here as they contribute the pixel-shifting $\boldsymbol{\mathcal{T}}$, and thus may be separated according to [2].

Hence, the phase distortions caused by atmospheric turbulence can be further described by a random vector $\mathbf{a}_{\mathbf{x}} = [a_{\mathbf{x},1}, a_{\mathbf{x},2}, a_{\mathbf{x},3}, \ldots]^T$ at each pixel $\mathbf{x}$ in an image, which forms a set of random fields [5]. As stated by Noll [11], each vector is a 0-mean Gaussian vector with a specified covariance matrix,

$$\mathbb{E}[\mathbf{a}_{\mathbf{x}}\mathbf{a}_{\mathbf{x}}^T] = \mathbf{R}. \quad (2)$$

Noll used the Zernike polynomials to describe the phase distortions resulting from a point source, resulting in the basis representation:

$$\phi_{\mathbf{x}}(R\boldsymbol{\rho}) = \sum_i a_{\mathbf{x},i}\mathbf{Z}_i(\boldsymbol{\rho}), \quad (3)$$

where $\boldsymbol{\rho}$ is a vector defined over the unit circle, and R is the radius of the imaging system's aperture.

This concept has been generalized to include *separate* positions $\mathbf{x}$ and $\mathbf{x}'$, which form a covariance tensor $\mathbb{E}[a_{\mathbf{x},i}a_{\mathbf{x}',j}]$. [5] states that one may quickly generate the turbulent distortions for an image of size $H \times W$, within suitable approximation, from these components in the following way:

1. For $i \in \{1, 2, \ldots, 36\}$, compute the power spectral density (PSD) $\mathbf{S}_i$ for each covariance function through the use of the Wiener–Khinchin theorem, $\boldsymbol{S}_i = \mathcal{F}\{\mathbb{E}[a_{\mathbf{x},i}a_{\mathbf{x}',j}]\}$, where $\mathcal{F}$ denotes the Fourier transform.
2. Generate 36 zero-mean unit variance random fields according to the covariance function $\mathbb{E}[a_{\mathbf{x},i}a_{\mathbf{x}',j}]$. This is done according to FFT-based methods, which use a complex white noise seed $\mathbf{n}$ to form a field $\mathbf{v}_i$ in the following way: $\mathbf{v}_i = \text{real}(\mathcal{F}^{-1}\{\sqrt{\mathbf{S}_i}\mathbf{n}\})$.
3. Perform a Cholesky decomposition of the matrix derived by Noll $\mathbf{R} = \mathbf{L}\mathbf{L}^t$, which in our case is of size 36×36. Denoting the concatenated fields as $\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{36}]^T$ with dimensions $36\times H\times W$, the final output random fields may be generated as $\mathbf{a}' = \mathbf{L}\mathbf{v}$.

4. Provide the Zernike coefficient fields $\mathbf{a}'$ to the Phase-to-Space transformation (P2S) to compute the PSF-basis coefficients $\boldsymbol{\beta}_{\mathbf{x}} = \mathcal{P}(\mathbf{a}'_{\mathbf{x},\{i\geq 4\}})$.
5. Apply the image warping followed by the spatially varying blur by the P2S coefficients as described in the main body of the paper.

For color images, the same process is carried out, with the spatially varying convolution occurring in the same way for all color channels in accordance with [9].

Although from a high level, the simulation process in this work is identical to that of [5], there are some critical differences:

1. The spatially varying convolution is modified to match the image formation process more accurately. Though this is detailed in the paper, we provide additional evidence of the importance of this modification in a later subsection of the supplementary document. This affects step (5) of the simulation.
2. We use a reformulated expression $\mathbb{E}[a_{\mathbf{x},i}a_{\mathbf{x}',j}]$ according to [3], which we detail in the next two subsections. This reformulation leads to an exact solution rather than the approximate solution of [4]. This primarily affects step (1) of the simulation process.
3. We modify the P2S basis functions to be resizable according to the camera and environmental constraints. This is done through a larger PSF training dataset which alleviates the aliasing from the previously generated set. The new P2S bases can vary from a large PSF (size 200×200 or more) down to accurately modeling a delta function. This affects steps (4) and (5) of the described process.

2.2. Spatially varying convolution re-formulation

The physical meaning of a PSF is the way in which a point *spreads* across the sensor plane, which we refer to as a *scattering* process. However, previous implementations of the P2S transform operate as a *gathering* process. If the PSF is spatially invariant, the difference is trivial, equivalent to the difference between correlation and convolution. In the spatially varying case, the difference is no longer negligible. The *gathering* process of previous simulators [4, 5, 9] can be written as

$$\boldsymbol{O} \approx \sum_{k=1}^{100} \beta_{\mathbf{x},k} \left[\boldsymbol{\psi}_k \circledast \boldsymbol{\mathcal{T}}(\boldsymbol{I})\right] + \boldsymbol{n}. \tag{4}$$

The *scattering* process is instead written as [15]:

$$\boldsymbol{O} \approx \sum_{k=1}^{100} \boldsymbol{\psi}_k \circledast \left[\beta_{\mathbf{x},k} \boldsymbol{\mathcal{T}}(\boldsymbol{I})\right] + \boldsymbol{n}. \tag{5}$$

While mathematically subtle, the difference is significant. Under the *gathering* model, a single point source at $\mathbf{x}_0$ (i.e. $\boldsymbol{\mathcal{T}}(\boldsymbol{I}) = \delta(\mathbf{x}-\mathbf{x}_0)$) will have the corresponding blur:

$$\boldsymbol{O} \approx \sum_{k=1}^{100} \boldsymbol{\psi}_k(\mathbf{x}-\mathbf{x}_0)\boldsymbol{\beta}_{\mathbf{x},k} + \boldsymbol{n}, \tag{6}$$

whereas the *scattering* model (4) results in

$$\boldsymbol{O} \approx \sum_{k=1}^{100} \boldsymbol{\psi}_k(\mathbf{x}-\mathbf{x}_0)\boldsymbol{\beta}_{\mathbf{x}_0,k} + \boldsymbol{n}. \tag{7}$$

We see (7) as a shifted basis representation, whereas (6) is a shifted basis with weights varying across the area of the PSF – a mismatch to the image formation process.

2.3. Varying C_n^2 path

While on the surface, the problem may seem solved as described by the simulation overview. There exist some issues both at the theoretical and practical levels. The later iterations of the Zernike-based simulations [5, 9] seek to rectify the practical limitations, though a key theoretical issue has remained. This leads us to introduce the two key fundamental limitations of the multi-aperture simulation:

1. **Approximate solution.** Within [4], a Taylor series is utilized to determine the correlation of the Zernike coefficients. This results in the solution only being approximate, unable to match the theoretical curves exactly as their approach utilizes a first-order Taylor approximation.
2. **Restriction to constant C_n^2-paths.** Related to the Taylor series is the inability to model any turbulence beyond ground-to-ground. Furthermore, ground-to-ground situations exist for which there is a non-trivial error by the approximation, along with the potential of heat sources along the path of propagation, which would make a constant turbulence strength assumption invalid.

These issues have been addressed by a recent analysis [3]. While it is primarily the subject of the mentioned paper, we feel it important to describe it to a sufficient level of detail here, as it is a critical improvement to the simulation quality which allows us greater accuracy in our simulations. That being said, we do not anticipate the reader who is unfamiliar with the atmospheric turbulence literature to understand the following set of equations. Therefore, we briefly present the main results for completeness and then offer an interpretation of the equations that do not require so much background.

As a wave propagates through a turbulent path, the strength of the turbulence, C_n^2, may vary along the propagation path. This motivates writing the strength as a function of propagation distance, $C_n^2(z)$. The new theoretical Zernike correlation result [3] allows one to write the autocorrelation of Zernike coefficients $\mathbb{E}[a_{\mathbf{x},i}a_{\mathbf{x}',j}]$ as a function of this continuous C_n^2-profile:

$$\mathbb{E} = \mathcal{A}_{i,j} \int_0^L \left(\frac{L-z}{L}\right)^{5/3} C_n^2(z) f_{ij}\left(vs.(z), k_0\right) dz \quad (8)$$

where $\mathcal{A}_{i,j} = 0.00969 k^2 2^{14/3} \pi^{2/3} R^{5/3} \sqrt{(n_i+1)(n_j+1)}$ and L is the length of propagation. The f_{ij} expression is provided by [13]: for a displacement $\mathbf{s} = (s, \varphi)$ written in polar form, the expression in [13] is written as

$$\begin{aligned} f_{ij}(vs., k_0) = (-1)^{(n^+ - m^+)/2} &\Theta^{(1)}(i,j) \\ &\times I_{m^+, n_i+1, n_j+1}(2s, 2\pi R k_0) \\ +(-1)^{(n^+ + 2m_i + |m^-|)/2} &\Theta^{(2)}(i,j) \\ &\times I_{|m^-|, n_i+1, n_j+1}(2s, 2\pi R k_0), \end{aligned} \quad (9)$$

with functions

$$I_{a,b,c}(s, k_0) = \int dx \frac{J_a(sx) J_b(x) J_c(x)}{x(x^2 + k_0)^2}, \quad (10)$$

along with angular functions

$$\Theta^{(1)}(i,j) = \begin{cases} (-1)^j \cos(m^+\varphi) & h(i,j) = 1 \\ \sin(m^+\varphi) & h(i,j) = 2 \\ \sqrt{2}\cos(m^+\varphi) & h(i,j) = 3 \\ \sqrt{2}\sin(m^+\varphi) & h(i,j) = 4 \\ 1 & h(i,j) = 5 \end{cases} \quad (11)$$

and,

$$\Theta^{(2)}(i,j) = \begin{cases} \cos(m^-\varphi) & h(i,j) = 1 \\ \sin(m^-\varphi) & h(i,j) = 2 \\ 0 & h(i,j) = 3 \\ 0 & h(i,j) = 4 \\ 0 & h(i,j) = 5 \end{cases}, \quad (12)$$

contributing the angular terms and

$$n^\pm = n_i \pm n_j, \quad (13)$$

$$m^\pm = m_i \pm m_j. \quad (14)$$

Though the equations which (8) utilizes are indeed tedious to write and interpret, (8) itself can be understood in a fairly straightforward manner. First, recall that $C_n^2(z)$ is the strength of the turbulent fluctuations. Thus, the correlation of the Zernike coefficients is a weighted summation of the turbulent distortions. The term $(L - z/L)^{5/3}$ says that turbulence *closer* to the camera contributes higher strength and longer correlation length than turbulence far away from the camera. The term $f_{ij}(\cdot)$ is a result of using the Zernike polynomials – therefore, it is simply a function that falls out of the mathematical description of them. The inner term $vs.(z)$ is a function of geometry, which ensures neighboring points have a higher correlation than points that are far apart. Finally, although k_0 is not so straightforward to interpret without proper background in the literature, it is related to the size of the turbulent distortions (not strength, but their geometric size).

We claim that (8) is a significant improvement over previous results of [4]. To demonstrate this difference, we use an example as given in [3] to show that the general result (8) contains the results of [4] as a special case. We offer some additional interpretation here to aid in understanding.

For this example, the turbulence strength is defined to be the following

$$C_n^2(z) = L C_n^2 \delta\left(z - \frac{L}{2}\right). \quad (15)$$

This means the turbulence is located at the halfway point of propagation, the rest is free space. If we plug this $C_n^2(z)$ function into (8), we achieve the same correlation function as in [4]:

$$\mathbb{E}[a_{\mathbf{x},i} a_{\mathbf{x}',j}; 1] = \mathcal{A}_{i,j} \left(\frac{1}{2}\right)^{5/3} L C_n^2 f_{ij}\left(\frac{(\mathbf{x} - \mathbf{x}')}{D}, k_0\right). \quad (16)$$

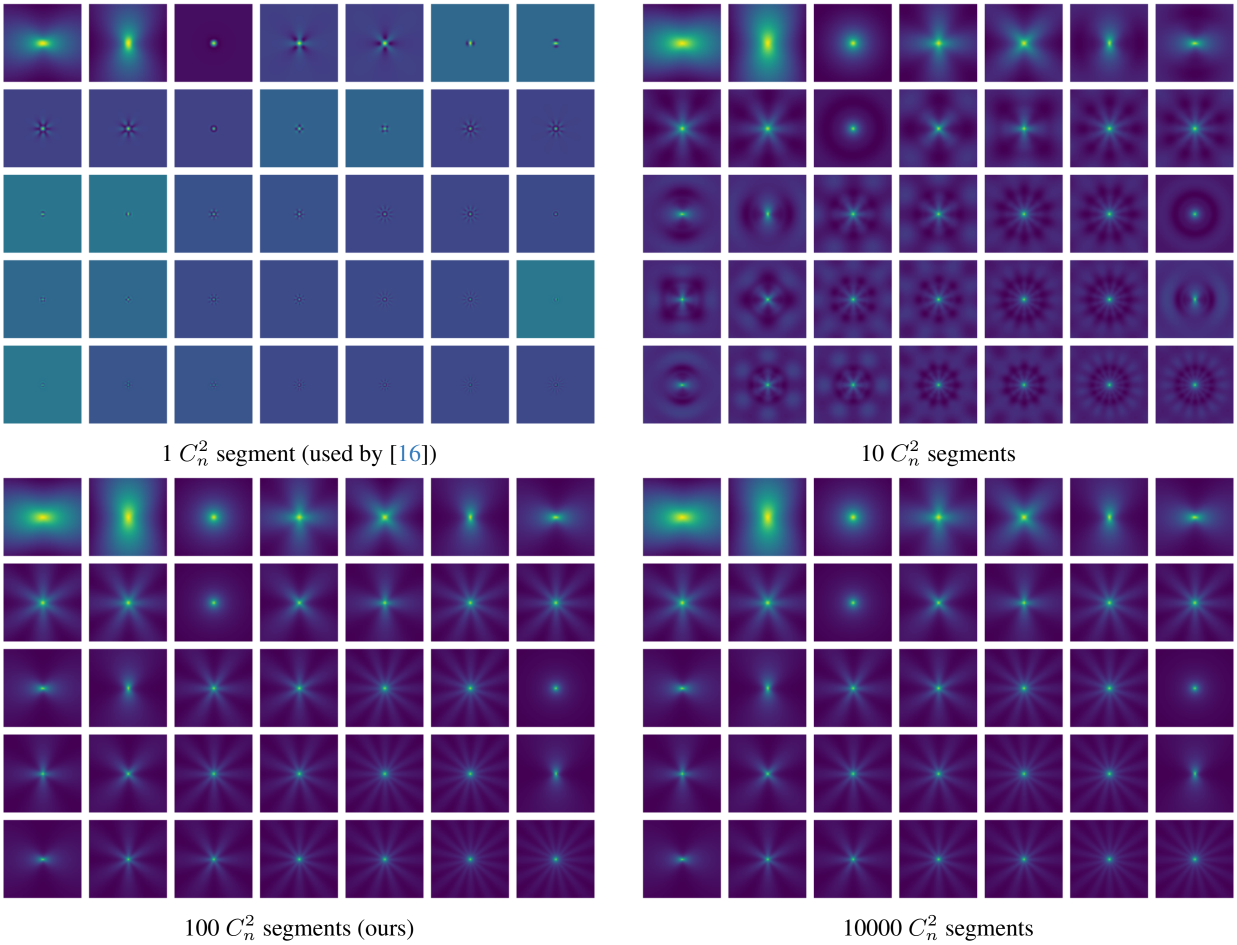
Interpreting this result means that previous Zernike-based simulations were equivalent to “squeezing” all of the turbulence into a single infinitesimally thin slice at the halfway point. This explains the inaccuracy by [4] as to why they cannot (i) exactly match theoretical predictions and (ii) be extended to varying C_n^2-profiles. Unknown to [4], their approximation is equivalent to approximating the integral of (8) as a single Riemann summation term.

Our approach to simulation in this paper rests on the result of [3], which is exact. Furthermore, it does not increase time in simulation, except for a small increase in precomputation, which has been suitably optimized. We note that this precomputation happens *once ever* as long as k_0 doesn’t change (which is not too restrictive of an assumption).

To visualize the improvement in this correlation term by the number of terms used to approximate the integral (8), we present a visualization in Figure 7. This demonstrates that (i) a few additional terms contribute a great deal to the overall accuracy and (ii) an increase in terms *decreases* the aliasing. The decrease in aliasing is because FFT-based generation is utilized – any high-frequency content, which is “blurred” out by additional terms, may be aliased if the sample grid is not large enough spatially. (iii) Our experiments demonstrate 10-100 phase points in evaluating (8) to be sufficient, depending on the situation.

2.4. New P2S kernels

In an optical simulation, careful consideration of the various sample spacings is critical for achieving high accuracy. Previous multi-aperture simulations have made some progress



1 C_n^2 segment (used by [16])

10 C_n^2 segments

100 C_n^2 segments (ours)

10000 C_n^2 segments

Figure 7. An instance of $\mathbb{E}[a_{\mathbf{x},i}a_{\mathbf{x}',i}]$ from (8) under different number of C_n^2 segments along the optical path. Here, we show the 2nd to 36th autocovariance functions in raster order, and brighter pixels indicate larger values. The associated parameter set is distance = 600m, focal length = 500mm, F-number = 11, $C_n^2[z] = 5 \times 10^{-14}\text{m}^{-2/3}$ for all z, image size = 128×128, scene width = 0.5m. From this figure, we find that the additional precision becomes negligible when we use more than 100 segments. Hence we chose 100 segments for data synthesis.

in this direction. However, their approach is limited in many ways. The reason for this reduces to the fact that their kernels ψ_i may not be easily resized. This hurts the accuracy of the simulation by causing mismatches in sampling and limits the model's generalizability.

The P2S kernels implemented in this paper are (i) resizable and (ii) chosen to match the sampling parameters of the scene. The core solution is (i), with (ii) being an important consequence of this correction. The main limitation in the P2S bases is their initial size of 33×33. This causes the bases too often to be aliased significantly upon resizing. To address this issue, we have increased the resolution of the PSF dictionary, resulting in the basis functions being of size 67×67. Additionally, the dictionary is 20× larger than [9], aiding in the eigenfunctions being well-behaved. The dictionary is generated with turbulence strength $D/r_0 = [0.1, 12]$, representing various turbulent conditions. Through our testing, we have observed we can match PSFs from a delta function up to the challenging cases of $6 \leq D/r0 \leq 12$.

With these modifications, we have observed no notable aliasing when resizing the PSF basis functions. This allows us to resize the bases to match the sampling specified in the simulation parameters. This is done by a tuning step that operates in the following way:

1. The basis is used to represent the diffraction kernel offline. We can compute the full width at half maximum (FWHM) in pixels for the basis N_d. This step is done once and hard-coded into the simulation.
2. Given the specified image size and camera parameters, the diffraction kernel FWHM can be computed in meters and converted to pixels N_0. This is done for every new

Modality	Distance (m)	Focal length (m)	F-number	Scene width (m)	$C_n^2(10^{-14} \times \mathrm{m}^{-2/3})$
ATSyn-dynamic	[30, 100]	[0.1, 0.3]	$\{2.8, 4\}$	[2, 4]	[50, 300]
			$\{2.8, 4, 5.6\}$	[4, 20]	[200, 1000]
	[100, 200]	[0.2, 0.5]	$\{2.8, 4, 5.6\}$	[2, 4]	[5, 50]
			$\{2.8, 4, 5.6\}$	[4, 20]	[20, 100]
	[200, 400]	[0.3, 0.5]	$\{5.6, 8\}$	[2, 6]	[2, 30]
			$\{4, 5.6, 8\}$	[6, 20]	[10, 40]
	[400, 600]	[0.4, 0.75]	$\{8, 11\}$	[3, 7]	[1, 20]
			$\{5.6, 8, 11\}$	[7, 20]	[10, 30]
	[600, 800]	[0.6, 0.8]	$\{8, 11\}$	[4, 8]	[1, 15]
			$\{8, 11\}$	[8, 20]	[2, 20]
	[800, 1000]	[0.8, 1]	$\{11, 16\}$	[4, 8]	[0.5, 10]
			$\{8, 11, 16\}$	[8, 20]	[1, 20]
ATSyn-static	[200, 400]	[1, 2]	$\{8, 11\}$	[0.2, 0.5]	[3, 7]
			$\{5.6, 8, 11\}$	[0.5, 1]	[6, 30]
	[400, 600]	[1, 2.5]	$\{8, 11, 16\}$	[0.4, 0.8]	[2, 6]
			$\{5.6, 8, 11\}$	[0.8, 1.5]	[6, 30]
	[600, 800]	[1, 3]	$\{11, 16\}$	[0.5, 1.2]	[2, 5]
			$\{8, 11\}$	[1.2, 2]	[5, 30]

Table 2. Parameter range, where $[a, b]$ means uniform sampling from continuous range (a, b), and $\{\}$ indicates uniform sampling from the discrete set, all rows were chosen with identical probability

Blur Strength	$k_b \leq 17$	$19 \leq k_b \leq 29$			$k_b \geq 31$
		$D/r_0 < 2$	$2 \leq D/r_0 \leq 8$	$D/r_0 > 8$	
Weak	$\bar{d} < 0.5$		$\bar{d} < 0.2$	-	
Medium	$0.5 \leq \bar{d} \leq 1$		$0.2 \leq \bar{d} \leq 0.4$	$\bar{d} \leq 0.2$	
Strong	$\bar{d} > 1$		$\bar{d} > 0.4$	$\bar{d} > 0.2$	

Table 3. Turbulence strength criterion in ATSyn-dynamic, the value of k_b is odd.

set of parameters.

3. The basis is resized by N_0/N_d, making the FWHM of the diffraction kernel coincide with the theoretically predicted value.

Through this process, we can correctly incorporate the sampling of the imaging system and scene into the basis representation. In addition, we optionally incorporate PSF basis size scaling by D/r_0. We have observed that this gives us additional turbulence blur not captured in the above PSF resizing scheme.

2.5. Temporal correlation

Real-world turbulence is temporally correlated because the dynamics of the atmosphere is a continuous process. Therefore, accurately simulating a video will require the degradation to be spatiotemporally correlated. We disentangled the spatial and temporal correlation and injected temporal correlation into the simulation process by correlating the initial random seed in the simulation. We use an $AR(1)$ process to generate the initial seed at the first stage. This allows for the random seed $\boldsymbol{n}_t$ at time t, which is then used to form the distortion and blur random fields, to be related to the previous realization by

$$\boldsymbol{n}_t = \alpha \boldsymbol{n}_{t-1} + \sqrt{1 - \alpha^2} \boldsymbol{\epsilon}_t \tag{17}$$

The term α is the temporal correlation ratio and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \boldsymbol{I})$.

3. ATSyn Dataset

The ATSyn dataset has two subsets: *ATSyn-dynamic* and *ATSyn-static*. The objective of the static scene turbulence mitigation task is to restore a single common ground truth from a sequence of degraded frames, which has been extensively explored in classical turbulence mitigation literature. On the other hand, the dynamic scene turbulence mitigation task aims to restore each video frame where the object or scene is in motion, presenting a significantly greater challenge for conventional methods. As stated in the main paper, the *ATSyn-dynamic* contains 5447 groups of turbulence-affected videos, the $\mathcal{T}$-only videos and ground truth videos. Among all 5447 groups, 4350 are for training, and 1097 are for validation. Frame-wise, we have 1816375 frame groups for training. We use the first 120 frames in each testing video during testing if the original testing video has more than 120 frames. On the other hand, the ATSyn-static subset contains 3000 groups of image sequences, each consisting of 50 turbulence-affected frames, 50 $\mathcal{T}$-only frames, and a corresponding ground truth image. Out of these 3000 groups, 2000 are designated for training, while 1000 are set aside for validation. Thanks to the efficiency of our simulator, the entire synthesis process can be completed within seven days using a single RTX 2080Ti GPU or 42 hours using a single NVIDIA A100 GPU.

3.1. Parameter selection details

Using the simulation method in Section 1, we can synthesize long-range atmospheric turbulence effects at various physical and camera parameters. These parameters include distance, the field of view (FOV) represented by scene width, turbulence profile indicator C_n^2, focal length, and F-number of the camera. The detailed parameter ranges are shown in Table 2. When setting the parameters, we first select the distance, FOV, focal length, and f-number with parameters ranging from a standard camera and lens to an astronomical telescope. We then choose the C_n^2 range to set the turbulence effect to be neither too strong nor weak. The temporal correlation was sampled from $0.2\sim0.9$ in the *ATSyn-static* and $0.3\sim0.95$ in the *ATSyn-dynamic*.

3.2. Turbulence strength

We classify the turbulence strength into multiple levels to study how turbulence mitigation networks perform under different conditions. For the ATSyn-dynamic dataset, we select three levels. Although our parameters are carefully chosen, the relationship between turbulence strength and parameters is highly nonlinear. We, therefore, determined the turbulence strength based on the actual degradation of the image. Turbulence degradation consists of the pixel displacement and blur effect. The average pixel displacement (denoted by $\overline{d}$) can measure the former. The latter can be indicated by the size of the blur kernel basis (denoted by k_b) and the turbulence strength D/r_0. The size of the blur kernel basis is related to, though not proportional to, D/r_0; the blur kernel size is also affected by the image resolution, distance, and field of view. It is possible that the same blur kernel basis yields different blur effects under different D/r_0 or that the same D/r_0 is associated with different blur sizes because the resolution of the blur kernel varies. Therefore, we need to consider both the size of the basis and D/r_0. The detailed classification criterion is listed in Table 3.

We use 4500 clean input videos to generate the dataset, partitioned into three groups with 1500 videos per partition. For each video, we run the parameter generator in Section 3.1 to produce random turbulence parameters and synthesize a single sample frame. The turbulence strength can be determined from this instance according to Table 3. We synthesize the entire video if the associated turbulence strength set is not full, or we abandon the set of parameters and randomly produce another set and repeat the steps above until the video is accepted by one turbulence strength set or all videos are synthesized.

References

[1] Nantheera Anantrasirichai. Atmospheric turbulence removal with complex-valued convolutional neural network. *Pattern Recognition Letters*, 171:69–75, 2023. 1, 3

[2] Stanley H. Chan. Tilt-then-blur or blur-then-tilt? clarifying the atmospheric turbulence model. *IEEE Signal Processing Letters*, 29:1833–1837, 2022. 4

[3] Nicholas Chimitt and Stanley Chan. Anisoplanatic optical turbulence simulation for near-continuous Cn2 profiles without wave propagation. *Optical Engineering*, 62(7):078103, 2023. 5, 6

[4] Nicholas Chimitt and Stanley H. Chan. Simulating anisoplanatic turbulence by sampling intermodal and spatially correlated Zernike coefficients. *Optical Engineering*, 59(8): 083101, 2020. 4, 5, 6

[5] Nicholas Chimitt, Xingguang Zhang, Zhiyuan Mao, and Stanley H Chan. Real-time dense field phase-to-space simulation of imaging through atmospheric turbulence. *IEEE Transactions on Computational Imaging*, 2022. 4, 5

[6] Jérôme Gilles and Nicholas B Ferrante. Open turbulent image set (OTIS). *Pattern Recognition Letters*, 86:38 – 41, 2017. 2

[7] D. Jin, Y. Chen, Y. Lu, J. Chen, P. Wang, Z. Liu, S. Guo, and X. Bai. Neutralizing the impact of atmospheric turbulence on complex scene imaging via deep learning. *Nature Machine Intelligence*, 3:876 – 884, 2021. 1, 3

[8] Nianyi Li, Simron Thapa, Cameron Whyte, Albert W. Reed, Suren Jayasuriya, and Jinwei Ye. Unsupervised non-rigid image distortion removal via grid deformation. In *IEEE/CVF International Conference on Computer Vision*, pages 2522 – 2532, 2021. 1

[9] Z. Mao, N. Chimitt, and S. H. Chan. Accelerating atmospheric turbulence simulation via learned phase-to-space transform. In *IEEE/CVF International Conference on Computer Vision*, pages 14759 – 14768, 2021. 5, 7

[10] N. G. Nair and V. M. Patel. Confidence guided network for atmospheric turbulence mitigation. In *IEEE International Conference on Image Processing*, pages 1359 – 1363, 2021. 1, 2

[11] R. J. Noll. Zernike polynomials and atmospheric turbulence. *Journal of Optical Society of America*, 66(3):207 – 211, 1976. 4

[12] Endre Repasi and Robert Weiss. Computer simulation of image degradations by atmospheric turbulence for horizontal views. In *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXII*, page 80140U. International Society for Optics and Photonics, 2011. 1

[13] N. Takato and I. Yamaguchi. Spatial correlation of Zernike phase-expansion coefficients for atmospheric turbulence with finite outer scale. *Journal of Optical Society of America A*, 12(5):958 – 963, 1995. 6

[14] UG2+. Bridging the gap between computational photography and visual recognition: $5th$ UG2+ prize challenge. http://cvpr2022.ug2challenge.org/dataset22_t3.html, 2022. Track 3. 2

[15] Kyrollos Yanny, Kristina Monakhova, Richard W. Shuai, and Laura Waller. Deep learning for fast spatially varying deconvolution. *Optica*, 9(1):96–99, 2022. 5

[16] Xingguang Zhang, Zhiyuan Mao, Nicholas Chimitt, and Stanley H. Chan. Imaging through the atmosphere using turbulence mitigation transformer. *IEEE Transactions on Computational Imaging*, 10:115–128, 2024. 7

[17] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 3