



SCHWEIZER JUGEND FORSCHT  
SCIENCE ET JEUNESSE  
SCIENZA E GIOVENTÙ

**KANTONSSCHULE  
ZOFINGEN**

**u<sup>b</sup>**

b  
**UNIVERSITÄT  
BERN**

**CSH**  
CENTER FOR SPACE AND  
HABITABILITY

# Entdeckung und Charakterisierung des Exoplaneten TOI 7265.01 mittels Transitmethode

Prämierte Maturaarbeit

von

Nikola Knežević

**Abbildung 1:** Schematische Darstellung eines Hot Jupiters. (Bildquelle: ESO/M. Kornmesser)



**Betreuende Person:** Dr. Erik Meier Valdés

**Betreuende Lehrperson:** Michael Franck

**Externe betreuende Personen:** Prof. Dr. Brice-Olivier Demory  
Doktorand Francis Zong Lang  
und die restliche Saint-Ex-Forschungsgruppe

**Eingereicht am:** 29. März 2026

# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>5</b>
<b>2</b>	<b>Abstract</b>	<b>6</b>
<b>3</b>	<b>Einleitung</b>	<b>6</b>
<b>4</b>	<b>Grundlagen</b>	<b>7</b>
4.1	Exoplaneten . . . . .	7
4.1.1	Exoplanetentypen . . . . .	8
4.2	Exoplanetkandidaten und TOIs . . . . .	8
4.3	Methoden zur Entdeckung von Exoplaneten . . . . .	9
4.3.1	Transitmethode . . . . .	9
4.3.2	Radialgeschwindigkeitsmethode . . . . .	11
4.3.3	Direkte Bildgebung . . . . .	11
4.4	Saint-Ex-Observatorium . . . . .	12
4.5	Der Transiting Exoplanet Survey Satellite . . . . .	13
4.6	Grundlagen des MCMC-Verfahrens . . . . .	14
4.7	Verwendete Software und Analysewerkzeuge . . . . .	14
4.7.1	Python . . . . .	15
4.7.2	Anaconda . . . . .	15
4.7.3	JupyterLabs . . . . .	15
4.7.4	BATMAN . . . . .	15
4.7.5	Emcee . . . . .	15
4.7.6	ExoFOP . . . . .	16
4.7.7	Swarthmore College Transit Finder . . . . .	16
4.7.8	AstrolmageJ . . . . .	16
4.8	Lyceum 130 Observatorium, Nowosibirsk . . . . .	16
4.9	NuanceGP . . . . .	18
4.10	Reparametrisierungen für effizientes MCMC-Sampling . . . . .	18
4.10.1	Espinoza (2018): $(r_1, r_2) \rightarrow (b, p)$ . . . . .	18
4.10.2	Kipping (2013): $(q_1, q_2) \rightarrow (u_1, u_2)$ . . . . .	19
4.10.3	Sozzetti et al. (2007): $\rho_\star \rightarrow a/R_\star$ . . . . .	20

4.10.4	Logarithmische Parameter . . . . .	20
<b>5</b>	<b>Methodik</b>	<b>21</b>
5.1	Auswahl des Exoplanetkandidaten . . . . .	21
5.2	Beobachtungsplanung . . . . .	21
5.2.1	Berechnung der Transitzeiten und Bestimmung des Beobachtungsziels .	21
5.3	Datenerhebung . . . . .	22
5.4	Datenreduktion und Bildkalibrierung . . . . .	23
5.4.1	Dark-Frames . . . . .	23
5.4.2	Flat-Frames . . . . .	24
5.4.3	Bias-Frames . . . . .	25
5.4.4	Kalibrierte Bilder . . . . .	25
5.5	Photometrische Analyse . . . . .	26
5.5.1	Identifikation des Zielsterns im Bild . . . . .	26
5.5.2	Differenzialphotometrische Analyse mit AIJ . . . . .	27
5.6	Simulation des Transits in Python . . . . .	29
5.6.1	Importieren und Einlesen der Daten . . . . .	29
5.6.2	Festlegen der Startwerte für das Transitmodell . . . . .	30
5.6.3	Transitsimulation und Modellimplementierung . . . . .	31
5.6.4	Log-Wahrscheinlichkeit und Modellvalidierung . . . . .	32
5.6.5	MCMC-Sampling zur Parameterschätzung . . . . .	33
5.6.6	Parameter-Umwandlung in physikalische Grössen . . . . .	34
5.6.7	Visualisierung der Modell-Zeitreihe . . . . .	35
5.6.8	Transformation der MCMC-Samples und Erstellung des Corner Plots . . .	36
5.6.9	Ausgabe und Visualisierung der Ergebnisse . . . . .	37
5.7	Gemeinsamer Multi-Instrument Transit-Fit . . . . .	38
5.7.1	Parametervektor und Prior-Verteilungen . . . . .	38
5.7.2	Log-Likelihood-Funktion . . . . .	39
5.7.3	MCMC-Konfiguration und Codestruktur . . . . .	39
<b>6</b>	<b>Resultate und Diskussion</b>	<b>40</b>
6.1	Transitmodell und Residuenplot anhand der Eigenbeobachtungen . . . . .	40
6.2	Corner-Plot der Saint-Ex-Daten . . . . .	41
6.3	Corner-Plot der TESS-Daten . . . . .	42

6.4	Vergleich von Saint-Ex- und TESS-Daten . . . . .	44
6.5	Resultate des gemeinsamen Multi-Instrument-Fits . . . . .	45
6.5.1	Globale Transitparameter . . . . .	45
6.5.2	Instrumentenspezifische Radiusverhältnisse und Gütemasse . . . . .	45
6.5.3	Corner-Plot der Posterioren . . . . .	45
6.5.4	Phasengefaltete Lichtkurven und Transitmodell . . . . .	46
6.5.5	Posterioren-Verteilung der Radiusverhältnisse . . . . .	47
6.6	Scheiternde Statistische Validierung mittels TRICERATOPS . . . . .	48
6.6.1	Ergebnis der TRICERATOPS-Analyse . . . . .	49
6.6.2	Ursache: Verfälschung des Lichtflusses durch die TESS-Apertur . . . . .	49
6.6.3	Warum TRICERATOPS für diesen Kandidaten nicht anwendbar ist . . . . .	51
6.7	Charakterisierung von TOI 7265.01 . . . . .	51
6.7.1	Die Masse-Radius-Relation für Riesenplaneten . . . . .	51
6.7.2	Eingrenzung durch Vergleich mit ähnlichen Systemen . . . . .	52
6.7.3	Zusammenfassung der Massenabschätzung . . . . .	52
6.7.4	Schlussfolgerung: TOI-7265.01 als echter Planetenkandidat . . . . .	53
<b>7</b>	<b>Reflexion</b>	<b>54</b>
<b>8</b>	<b>Danksagung und Widmung</b>	<b>56</b>
<b>9</b>	<b>Referenzen</b>	<b>57</b>
<b>10</b>	<b>Abbildungsverzeichnis</b>	<b>60</b>
<b>11</b>	<b>Tabellenverzeichnis</b>	<b>62</b>
<b>12</b>	<b>Formelverzeichnis</b>	<b>63</b>
<b>13</b>	<b>Anhang</b>	<b>66</b>
13.1	Python-Quellcode des Joint-Fit-Notebooks . . . . .	66
13.2	Python-Code: TRICERATOPS-Validierung . . . . .	86
<b>14</b>	<b>Eigenständigkeitserklärung</b>	<b>90</b>

# 1 Vorwort

Bereits im Kindesalter begeisterte mich das Universum durch seine Dimensionen und Schönheit. Extrasolare Planeten faszinierten mich seit dem ersten Mal, als ich davon hörte, da für mich die Vorstellung anderer Welten ausserhalb der Erde so interessant ist. Obwohl ich bei der Themenwahl für meine Maturaarbeit zunächst skeptisch war, ob ich sie umsetzen könnte, wollte ich die Herausforderung annehmen.

Glücklicherweise erhielt ich die Gelegenheit, mit Forschenden der Universität Bern und dem Team des Saint-Ex-Observatoriums in San Pedro Mártir, Mexiko, zusammenzuarbeiten. Nach einer Einarbeitungsphase in die Grundlagen der Materie entschied ich mich, einen Exoplaneten photometrisch zu beobachten und auszuwerten. Das wichtigste Ziel für mich war, einen wissenschaftlichen Mehrwert zu erschaffen und dies ist mir mit der Hilfe von allen internen und externen Betreuungspersonen gelungen.

Die Motivation für diese Maturaarbeit war leicht zu finden, insbesondere weil die Beobachtung eines fernen Sterns und dessen Exoplaneten eine spannende Verbindung von Technologie und Weltraum darstellt. Diese Arbeit hat mir sehr viel Freude bereitet, da ich viele Menschen kennenlernen durfte, ich einen Einblick in aktuelle Forschungsgebiete erhalten habe und ich wertvolle Erfahrungen erlebt habe.

Während der Arbeit an diesem Projekt habe ich nicht nur mein technisches Verständnis für astronomische Messmethoden vertieft, sondern auch meine Fähigkeit zur Problemlösung und kritischem Denken verbessert. Die Zusammenarbeit mit internationalen Forschenden hat meine Kommunikationsfähigkeiten gestärkt und mir gezeigt, wie wichtig Teamarbeit in der Forschung ist.

Insgesamt hat mich dieses Projekt persönlich sehr wachsen lassen und meinen Wunsch, in der Forschung mitzuhelfen, gestärkt.

## 2 Abstract

Diese Maturaarbeit befasst sich mit der photometrischen Charakterisierung des Exoplanetenkandidaten TOI-7265.01, eines jupitergrossen Gasriesen im Orbit um den M-Zwerg TIC 275717567 ( $T_{\text{eff}} = 3392 \pm 200$  K,  $R_{\star} = 0,488 \pm 0,015 R_{\odot}$ ,  $M_{\star} = 0,486 \pm 0,021 M_{\odot}$ ). Die Analyse basiert auf einem gemeinsamen Transit-Fit dreier unabhängiger photometrischer Datensätze: Beobachtungen mit dem 1-m-Teleskop des Saint-EX-Observatorium in San Pedro Mártir in Mexiko (534 Datenpunkte), dem 0,28-m-Teleskop des Lyceum 130 Observatoriums (LHO) in Nowosibirsk in Russland (62 Datenpunkte) sowie Lichtkurven des TESS-Satelliten im Orbit um die Erde (11'766 Datenpunkte). Die TESS-Daten wurden vorgängig mit dem NuanceGP-Algorithmus von stellarer Variabilität und instrumentellen Systematiken bereinigt.

Der gemeinsame Fit verwendet ein 19-dimensionales Markov-Chain-Monte-Carlo-Verfahren (MCMC) mit dem emcee-Sampler (150 Walker, 50 000 Schritte, 65 % Burn-in) und dem analytischen Transitmodell batman. Die Reparametrisierung des Impact-Parameters und des Radiusverhältnisses nach Espinoza (2018) sowie der Limb-Darkening-Koeffizienten nach Kipping (2013) gewährleistet eine effiziente Abtastung des Parameterraums mit annähernd gauss'schen Posteriorverteilungen.

Die Ergebnisse zeigen einen Planeten mit einer Orbitalperiode von  $P = 4,1723 \pm 0,0001$  Tagen und einem Radius von  $R_p = 10,52 \pm 0,25 R_{\oplus} = 0,938 \pm 0,023 R_{\text{Jup}}$ . Die instrumentenspezifischen Radiusverhältnisse betragen  $R_p/R_s = 0,201 \pm 0,006$  (SaintEX),  $0,199 \pm 0,008$  (LHO) und  $0,180 \pm 0,004$  (TESS), wobei die leichte Abweichung des TESS-Wertes auf Flussverdünnung durch benachbarte Sterne in der TESS-Apertur zurückzuführen ist. Der Impact-Parameter beträgt  $b = 0,30 \pm 0,10$  und die Sterndichte  $\rho_{\star} = 5,89 \pm 0,47 \text{ g cm}^{-3}$ . Die Gleichgewichtstemperatur des Planeten wird auf  $T_{\text{eq}} = 565 \pm 6$  K geschätzt. Aus der Masse-Radius-Relation nach Chen & Kipping (2017) ergibt sich eine geschätzte Masse von  $M_p \approx 0,5 - 1,5 M_{\text{Jup}}$ , wobei die grosse Spannweite die intrinsische Entartung der Masse-Radius-Relation im jovianischen Regime widerspiegelt. Die reduzierten  $\chi^2_{\nu}$ -Werte liegen für alle Instrumente nahe 1 (1,01 bis 1,23), was eine gute Modellanpassung ohne Überanpassung bestätigt.

Die statistische Validierung mittels TRICERATOPS ergibt ein hohes FPP von 0,987, das auf die extreme Flussverdünnung in der TESS-Apertur zurückzuführen ist (Zielstern-Flussanteil lediglich 4,3 %). Dieses Ergebnis unterstreicht die Notwendigkeit bodengebundener Nachbeobachtungen und validiert den Ansatz dieser Arbeit. Die NEB-Analyse schloss 126 von 271 benachbarten Sternen als Kontaminationsquellen aus. Anhand aller stellaren und planetaren Parametern, welche in dieser Arbeit berechnet wurden, handelt es sich bei TOI-7265.01 höchstwahrscheinlich um einen *Warm Jupiter* um einen M-Kategorie-Zwergstern.

TOI-7265.01 wird als robuster Planetenkandidat klassifiziert. Mit einem Radius knapp unter dem von Jupiter und einer Orbitalperiode von rund vier Tagen um einen kühlen M-Zwerg gehört dieses System zu den seltenen Gasriesen um massearme Sterne (Vorkommenshäufigkeit  $< 1\%$ ), die die gängigen Planetenentstehungsmodelle vor erhebliche Herausforderungen stellen. Eine Radialgeschwindigkeitsbestätigung mit modernen Spektrographen (erwartete Halb-amplitude  $K \approx 100 - 300$  m/s) wäre der nächste Schritt zur endgültigen physikalischen Bestätigung in Addition zu der hier aufgeführten statistisch-basierten Validationsmethodik.

## 3 Einleitung

In dieser Arbeit wird die Identifikation und Charakterisierung eines grossen Exoplaneten untersucht, der einen kleinen, kühlen Stern umkreist. Diese Thematik ist von Bedeutung, da bisher

kein physikalisches Modell existiert, das die oben genannten Sternsystemumstände zufriedenstellend beschreibt.

Im Rahmen dieser Arbeit werden die physikalischen Parameter des Exoplaneten bestimmt und es wird überprüft, ob es sich bei dem betrachteten TOI (TESS Object of Interest) tatsächlich um einen Exoplaneten handelt oder um eine falsch-positive Beobachtung.

Die Leitfragen, nach welchen sich diese Arbeit richtet, lauten:

- Wie kann ein Exoplanetkandidat anhand von Daten des Saint-Ex-Observatoriums mithilfe selbst entwickelter Computerprogramme identifiziert und analysiert werden?
- Welche physikalischen Parameter des TOIs lassen sich bestimmen?
- Handelt es sich bei dem gewählten TOI um einen echten Exoplaneten oder um ein falsch-positives Signal?

Die geplanten Beobachtungszeitpunkte und -positionen wurden mittels eigener Rechenverfahren sowie des Transit Finders des Swarthmore Colleges und des TESS Follow-up Observing Programms (ExoFOP) berechnet. Die Beobachtung wurde von der Forschungsgruppe des Saint-Ex-Observatoriums in Mexiko durchgeführt. Die Datenbearbeitung und -auswertung erfolgte mit AstrolmageJ und diversen Python-Programmen.

Die externe Betreuung durch Prof. Dr. Brice-Olivier Demory und den Doktoranden Francis Zong Lang von der Universität Bern gewährleistet die Qualität und Plausibilität der Analyse.

Darüber hinaus trägt diese Arbeit wissenschaftlich zu einem Fortschritt bei, indem sie Erkenntnisse liefert, die für die Entwicklung eines physikalischen Modells zur Entstehung und Charakterisierung von Jupiter-grossen Exoplaneten um kleine (0.1-0.7 Sonnenradien), kühle (2'400K-3'700K) M-Kategorie-Sterne von Bedeutung sind.

## 4 Grundlagen

In diesem Kapitel werden die Grundlagen vermittelt, die zum Verständnis der Forschung an Exoplaneten und der angewandten Methoden erforderlich sind. Ziel ist es, den aktuellen Wissensstand und die zentralen Begrifflichkeiten verständlich darzulegen, um das folgende Vorgehen nachvollziehbar zu machen. Zudem werden die Instrumente und Analyseverfahren vorgestellt, die für die Datenerhebung und Auswertung der photometrischen Beobachtungen am Saint-Ex-Observatorium genutzt werden. Diese Grundlagen bilden die Basis für die Interpretation der Ergebnisse und die Beantwortung der Forschungsfragen.

### 4.1 Exoplaneten

Exoplaneten, auch extrasolare Planeten genannt, sind Planeten, die ausserhalb unseres Sonnensystems um andere Sterne kreisen (NASA, 2025a). Sie können sich stark von den Planeten in unserem Sonnensystem unterscheiden. Bisher haben Forschende 6'007 Exoplaneten entdeckt und bestätigt (NASA, 2025a). Ihre Erforschung ist essenziell, um die Entstehung von Sonnensystemen zu verstehen und um zu erforschen, unter welchen Voraussetzungen sich Leben formen könnte.

### 4.1.1 Exoplanetentypen

Bisherige Erforschungen haben mehrere Haupttypen von Exoplaneten identifiziert, die sich durch ihre Masse, ihren Radius und ihre Umlaufbahn charakterisieren lassen (Kopparapu et al., 2013; NASA, 2025b; University of Exeter, 2025). Die wichtigsten Kategorien sind:

- Gasriesen: Diese Planeten besitzen eine Masse und einen Radius, die meist grösser als das Vierfache des Erdradius und Erdmasse sind, und bestehen überwiegend aus Wasserstoff und Helium. Hot Jupiters sind besonders nah am Stern, haben Umlaufzeiten von wenigen Tagen und aufgeblähte Atmosphären (NASA, 2025b).
- Neptunartige Planeten: Sie sind kleiner als Gasriesen (Radius ca. 2-4 Erdradien), haben meist eine dicke Wasserstoff-Helium-Atmosphäre und können als „Mini-Neptunes“ auftreten (Fiveable Content Team, 2025).
- Supererden: Ihre Masse liegt zwischen einer und zehn Erdmassen ( $1 - 10 M_{\oplus}$ ), ihr Radius zwischen 1.25 und 2 Erdradien. Je nach Beschaffenheit können sie felsig (terrestrisch) oder teils eisreich sein (Fiveable Content Team, 2025; The Planetary Society, 2025).
- Terrestrische Planeten: Dies sind Gesteinsplaneten mit einer Masse und einem Radius ähnlich der Erde. Sie sind potenziell besonders interessant für die Suche nach Leben, da sie Bedingungen wie auf der Erde aufweisen können (Kopparapu et al., 2013).
- Weitere Spezialtypen: Zusätzlich gibt es noch exotische Gruppen wie „Rogue Planets“ (Planeten ohne Mutterstern), „Circumbinary Planets“ (Planeten, die um zwei Sterne kreisen), „Hycean Planets“ (wasserdominierte Exoplaneten) und erdgrosse Planeten in habitablen Zonen (NASA, 2025b; The Planetary Society, 2025; University of Exeter, 2025).

Die Vielfalt der bekannten Exoplaneten illustriert die Möglichkeiten planetarer Systeme nicht nur in ihrer Struktur, sondern auch hinsichtlich ihrer potenziellen Bewohnbarkeit und der Prozesse der Planetenentstehung im Kosmos.

## 4.2 Exoplanetkandidaten und TOIs

Ein Exoplanetenkandidat ist ein Himmelskörper, dessen Existenz aufgrund erster Messungen vermutet wird, welcher aber noch nicht endgültig bestätigt ist. Ein wichtiger Begriff hierbei ist das TOI (TESS Object of Interest). Ein TOI ist ein Exoplanetenkandidat, welcher von der NASA-Mission «Transiting Exoplanet Survey Satellite» (TESS) als potenzieller Exoplanet eingestuft wird und weitere Bestätigung benötigt. TESS überwacht Sterne auf periodische Helligkeitsabnahmen, sogenannte Transits, welche durch einen um den Stern kreisenden Exoplaneten verursacht werden können (Ricker et al., 2014).

Der Name eines TOIs besteht aus zwei Teilen. Die Zahl vor dem Punkt bezeichnet den Stern, der im Fokus der Beobachtung steht. Die Zahl nach dem Punkt gibt den potenziellen Exoplaneten an, welcher um diesen Stern kreist (Guerrero et al., 2021). Beispielsweise steht in «TOI 7265.01» die «7265» für den beobachteten Stern, während «.01» den ersten potenziellen Exoplaneten um diesen Stern kennzeichnet. Falls weitere Exoplaneten vermutet werden, würde die nachfolgende Dezimalstelle benutzt werden. Die Benennung folgt einem standardisierten System, um den Überblick über die Exoplanetenkandidaten zu erhalten.

## 4.3 Methoden zur Entdeckung von Exoplaneten

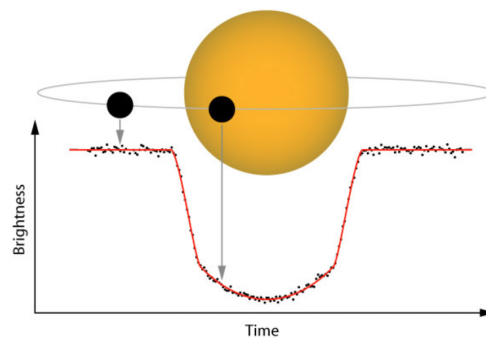
Es gibt unzählige Methoden zur Entdeckung eines Exoplaneten. Diese beruhen darauf, die Anwesenheit eines Exoplaneten durch seine indirekten oder direkten Einflüsse auf seinen Mutterstern oder auf das umgebende Licht nachzuweisen. Die Wahl der Methode hängt dabei von den Eigenschaften des jeweiligen Planetensystems und der verfügbaren Beobachtungstechnologie ab. Mit der Weiterentwicklung von Teleskopen und Datenanalyseverfahren hat die Wissenschaft eine Vielfalt an Verfahren etabliert, mit welchen es möglich ist, Exoplaneten zu charakterisieren. Details zu den wichtigsten Methoden werden in den folgenden Kapiteln erläutert.

### 4.3.1 Transitmethode

Die wichtigste Methode zur Entdeckung vieler Exoplaneten ist die Transitmethode. Diese wurde auch im Rahmen dieser Arbeit verwendet. Dabei wird die Helligkeit eines Sterns kontinuierlich gemessen. Zieht ein Planet vor dem Stern vorbei, wird die Helligkeit für kurze Zeit verringert. Durch weiteres Aufbessern der erlangten Daten kann ein Modell des Transits erstellt werden, sodass einige Parameter des Exoplaneten bestimmt werden können. Die durch das Modell berechneten Parameter umfassen den Planetenradius, die Umlaufzeit, die Transittiefe, die Transiddauer, den mittleren Transitzeitpunkt, die Bahnneigung anhand des Stossparameters, die mittlere Sterndichte, das Verhältnis des Planetenradius zum Sternradius und die beiden Randverdunkelungskoeffizienten.

#### Abbildung 2:

*Schematische Darstellung einer Lichtkurve beim Transit eines Exoplaneten vor seinem Mutterstern. (Bildquelle: NASA)*



Wie in Abbildung 2 zu sehen, tritt während eines Transits ein charakteristischer Abfall der Sternhelligkeit auf, der sich durch ein zeitlich begrenztes Muster auszeichnet. Die Analyse solcher Lichtkurven ermöglicht es, Planetenparameter zu bestimmen.

Im Folgenden werden alle Formeln für die Exoplanetparameter, welche durch die Transitmethode berechnet werden können, aufgelistet. So kann eine mathematische Grundlage für die späteren Analysen und Python-Simulationen garantiert werden.

**Formel 1:** Planetenradius (Winn, 2014)

$$R_p = R_s \times \sqrt{\delta}$$

**Formel 2:** Umlaufperiode (Seager & Mallén-Ornellas, 2003)

$$P = \frac{2\pi a^{3/2}}{\sqrt{GM_\star}}$$

**Formel 3:** Transittiefe (Winn, 2014)

$$\delta = \left(\frac{R_p}{R_s}\right)^2$$

**Formel 4:** Transitdauer (Seager & Mallén-Ornellas, 2003)

$$T_{14} = \frac{PR_s}{\pi a} \sqrt{\left(1 + \frac{R_p}{R_s}\right)^2 - b^2}$$

**Formel 5:** Mittlerer Transitzeitpunkt

$$t_0 = T_c \quad (\text{bestimmt aus der Lichtkurve})$$

**Formel 6:** Mittlere Sterndichte (Loeb & Gaudi, 2018)

$$\rho_\star = \frac{3\pi}{GP^2} \left(\frac{a}{R_s}\right)^3$$

**Formel 7:** Verhältnis Planetenradius zum Sternradius (Winn, 2014)

$$\frac{R_p}{R_s} = \sqrt{\delta}$$

**Formel 8:** Randverdunkelungs-Koeffizienten  $u_1, u_2$  (experimentell bestimmt, keine allgemeine Formel)

Planetenradius  $R_p$  ist der Radius des Exoplaneten, berechnet aus der Transittiefe relativ zum Sternradius  $R_s$ . Die Umlaufperiode  $P$  beschreibt die Zeit, welche der Planet benötigt, um einmal um seinen Stern zu kreisen. Die Transittiefe  $\delta$  gibt den relativen Helligkeitsabfall des Sterns an, wenn der Planet vor ihm vorbeizieht. Die Transitdauer  $T_{14}$  misst die Zeitspanne vom Eintritt bis zum Austritt des Planeten vor dem Stern. Der mittlere Transitzeitpunkt  $t_0$  ist das genaue Zeitfenster, zu dem der Planet die zentrale Sternscheibe vollständig überdeckt. Die Bahnneigung  $i$  beschreibt den Winkel zwischen der orbitalen Ebene des Planeten und der Sichtlinie zur Erde, basierend auf dem Stossparameter  $b$ , welcher den Abstand des Planetenpfads zur Sternmitte normiert. Die mittlere Sterndichte  $\rho_\star$  kann aus Keplerschem Gesetz und Transitgeometrie abgeleitet werden. Das Verhältnis  $\frac{R_p}{R_s}$  ist direkt aus der Quadratwurzel der Transittiefe berechenbar. Die Randverdunkelungs-Koeffizienten  $u_1$  und  $u_2$  beschreiben die abnehmende Helligkeit der Sternoberfläche zum Rand hin und werden später durch eine Simulation berechnet.

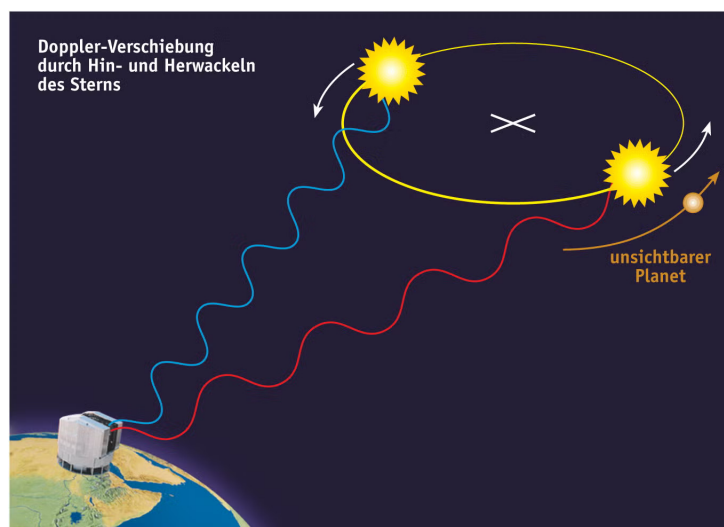
Für eine genauere Erklärung der Formeln können diese im Kapitel Formelverzeichnis nachgeschlagen werden.

### 4.3.2 Radialgeschwindigkeitsmethode

Die Radialgeschwindigkeitsmethode, auch Doppler-Spektroskopie genannt, war eine der ersten und ist bis heute eine der erfolgreichsten Techniken zur Entdeckung von Exoplaneten. Sie basiert darauf, dass Stern und Planet einen gemeinsamen Massenschwerpunkt umkreisen. Dadurch bewegt sich der Stern leicht auf und ab, was zu winzigen Dopplerverschiebungen in seinem Spektrum führt. Die Messung dieser Verschiebungen erlaubt Rückschlüsse auf die Masse des Planeten und die Eigenschaften seiner Bahn. Diese Methode eignet sich besonders gut für die Detektion von massereichen Planeten, da kleinere Planeten nur geringe Effekte erzeugen.

#### Abbildung 3:

*Illustration der Radialgeschwindigkeitsmethode: Darstellung der Bewegung von Stern und Planet sowie der Dopplerverschiebungen. (Bildquelle: SUW-Grafik; Erde: Terraforma)*



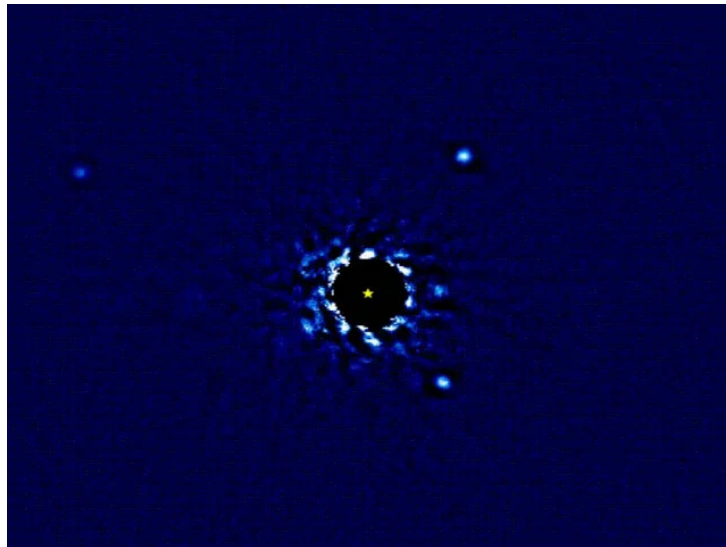
Durch kontinuierliche Beobachtungen und Datenanalysen ist es möglich, periodische Schwankungen in der Sternfrequenz zu erkennen und so die Existenz eines Planeten zu bestätigen. Insbesondere die Kombination dieser Methode mit anderen Techniken ermöglicht eine umfassende Charakterisierung von Exoplanetensystemen, etwa die Bestimmung von Masse, Umlaufbahn und sogar atmosphärischen Eigenschaften der Planeten. Die Radialgeschwindigkeitsmethode stellt damit einen Eckpfeiler der modernen Exoplanetenforschung dar.

### 4.3.3 Direkte Bildgebung

Die direkte Bildgebung ist eine technisch anspruchsvolle Methode, bei der versucht wird, Exoplaneten unmittelbar neben ihrem Mutterstern sichtbar zu machen. Dabei muss das viel hellere Sternenlicht mit speziellen Techniken stark unterdrückt werden, damit das lichtschwache Abbild des Planeten erkennbar wird. Diese Methode kommt vor allem bei grossen, jungen und relativ weit von ihrem Stern entfernten Planeten zum Einsatz und liefert wertvolle Informationen über deren Eigenschaften, insbesondere die Atmosphäre und Helligkeit.

#### **Abbildung 4:**

*Direkte Bildgebung eines extrasolaren Planeten. Die Aufnahme zeigt, wie ein Exoplanet im direkten Bild von seinem Mutterstern getrennt detektiert werden kann. (Bildquelle: Jason Wang et al., 2008)*



Mit Hilfe der direkten Bildgebung können Forschende nicht nur die Existenz von Exoplaneten nachweisen, sondern auch Rückschlüsse auf die chemische Zusammensetzung, Temperatur und mögliche Wetterphänomene ihrer Atmosphäre gewinnen. Die Methode erfordert modernste Teleskope sowie adaptive Optik und stellt eine wichtige Ergänzung zu den indirekten Nachweisverfahren dar.

#### **4.4 Saint-Ex-Observatorium**

Das Saint-Ex-Observatorium befindet sich in San Pedro Mártir, Mexiko. Es verfügt über ein vollautomatisches 1-Meter-Ritchey-Chrétien-Teleskop, das speziell für die Nachbeobachtung potenzieller Transit-Exoplaneten, wie etwa TOIs, konstruiert wurde (Demory, 2024). Saint-Ex ist Teil eines internationalen Forschungsnetzwerks, unter anderem mit der Universität Bern, und ergänzt durch Beobachtungen am Boden Weltraummissionen (Demory, 2024). Die Datenaufnahme und Auswertung erfolgen automatisiert durch eine eigens entwickelte Pipeline, die Lichtkurven erzeugt und systematische Messfehler korrigiert (Demory, 2024). Das Observatorium liegt auf einer Höhe von 2780 Metern und hat die geographischen Koordinaten  $31.04342^\circ$  N und  $115.45476^\circ$  W (Demory, 2024).

**Abbildung 5:**

Das Saint-Ex-Observatorium in San Pedro Mártir, Mexiko. (Bildquelle: Institute of Astronomy, UNAM / E. Cadena)



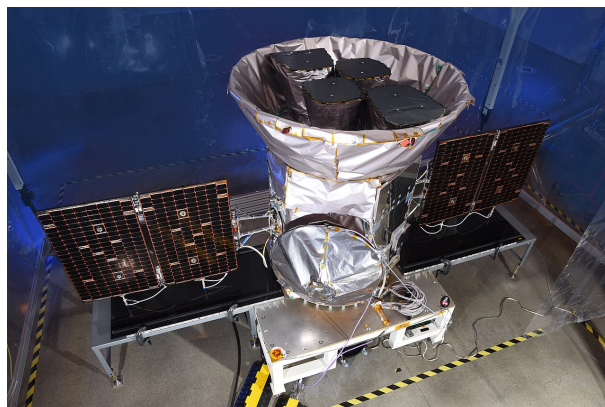
Die vollautomatische Steuerung, kombiniert mit einer Datenanalyse-Pipeline, garantiert eine effiziente und präzise Erfassung von Lichtkurven. Damit leistet das Saint-Ex-Observatorium einen wichtigen Beitrag zur Validierung von Exoplanetkandidaten und ergänzt die Beobachtungen von Weltraummissionen optimal. Das Observatorium kann durch den Breitbandfilter Wellenlängen von etwa 320nm bis 1000nm beobachten (Demory, 2024).

#### 4.5 Der Transiting Exoplanet Survey Satellite

Der *Transiting Exoplanet Survey Satellite* (TESS) ist ein NASA-Weltraumteleskop, das seit 2018 den gesamten Himmel systematisch nach Transitereignissen um helle, nahe Sterne durchsucht. TESS arbeitet mit vier Weitwinkelkameras, die gemeinsam ein Sichtfeld von  $24^\circ \times 96^\circ$  abdecken und damit in 27-tägigen Sektoren grosse Himmelsregionen kontinuierlich überwachen. Die Mission liefert hochpräzise, durchgängige Photometrie über Wochen bis Monate hinweg und bildet damit eine zentrale Grundlage für die Entdeckung und Ephemeriden-Bestimmung von Exoplaneten.

**Abbildung 6:**

Foto des TESS-Satelliten im Labor. (Orbital ATK / NASA)



TOI 7265 wurde ursprünglich in den TESS-Daten als Transitkandidat identifiziert. Die TESS-Lichtkurve lieferte eine erste Abschätzung der Umlaufperiode, Transitdauer und Transittiefe des

Planeten und ermöglichte die Bestimmung einer vorläufigen Ephemeride. Aufgrund des sektorbasierten Beobachtungsplans von TESS sowie der begrenzten Zeitabdeckung waren diese Parameter jedoch anfällig für systematische Unsicherheiten, insbesondere in der Transitmitte und Umlaufperiode. Diese Unsicherheiten beeinflussen direkt die Vorhersage zukünftiger Transite und damit die Planung von bodengebundenen Folgebeobachtungen.

In dieser Arbeit dienen die TESS-Daten als globaler Referenzanker im späteren Joint-Fit-Modell des Systems TOI 7265.01. Die Raumdaten liefern eine lange, homogene Zeitreihe mit hoher Signal-zu-Rausch-Ratio, während die bodengebundenen photometrischen Beobachtungen einzelne Transite mit höherer räumlicher Auflösung und besserer Kontrolle über Blendquellen ergänzen. Durch die Kombination der TESS-Lichtkurve mit den hier vorgestellten Bodenbeobachtungen in einem gemeinsamen Fit lassen sich die Transitparameter (Ephemeride, Tiefe, Dauer und Impact Parameter) konsistenter und mit kleineren Unsicherheiten bestimmen, als es mit einem der Datensätze allein möglich wäre. Insbesondere wird die Umlaufperiode von TOI 7265.01 durch den Joint Fit deutlich besser eingegrenzt, was zu präziseren Transitvorhersagen und somit zu effizienterer Planung zukünftiger Folgebeobachtungen führt.

## 4.6 Grundlagen des MCMC-Verfahrens

Das MCMC-Verfahren (Markov-Chain-Monte-Carlo-Verfahren) ist eine Methode zur effizienten Stichprobenziehung aus komplexen Wahrscheinlichkeitsverteilungen, die besonders in der Astrophysik und Exoplanetenforschung zur Parameterabschätzung von Modellen angewandt wird (Foreman-Mackey et al., 2013). Dabei wird eine Markov-Kette konstruiert, deren Zustände die Parameterkonfigurationen repräsentieren, sodass nach ausreichender Mischzeit die Verteilung der Zustände der gesuchten Posteriorverteilung entspricht.

Eine Markov-Kette ist ein mathematisches Modell für einen Prozess, bei dem der nächste Zustand nur vom aktuellen Zustand abhängt und nicht davon, wie der Zustand erreicht wurde. Diese „Gedächtnislosigkeit“ wird als Markov-Eigenschaft bezeichnet (Foreman-Mackey et al., 2013). Das bedeutet, dass die Übergangswahrscheinlichkeit zum nächsten Zustand nur von der aktuellen Position bestimmt wird. Mathematisch lässt sich dies über eine Übergangsmatrix darstellen, die alle Wahrscheinlichkeiten für Zustandswechsel enthält (Foreman-Mackey et al., 2013).

Ein populärer Algorithmus zur Umsetzung von MCMC ist der Metropolis-Hastings-Algorithmus (Hastings, 1969; Metropolis et al., 1953), welcher bei jedem Schritt einen neuen Kandidatenparameter vorschlägt und diesen anhand der Verhältnisse der Wahrscheinlichkeiten akzeptiert oder verwirft. Dies gewährleistet eine Monte-Carlo-basierte Integration der Posteriorverteilung und erlaubt damit eine vollständige statistische Charakterisierung der Parameter, inklusive Unsicherheiten und Korrelationen.

Insbesondere bei der Modellierung von Exoplanetentransits ist MCMC essentiell, da die Parameterräume oft hochdimensional und nicht trivial geformt sind. Durch MCMC können so robuste, probabilistische Aussagen über Stern- und Planetenparameter getroffen werden (Ford, 2005; Gregory, 2007).

## 4.7 Verwendete Software und Analysewerkzeuge

Für die Analyse und Auswertung photometrischer Messdaten kommen verschiedene spezialisierte Softwaretools zum Einsatz. Die folgenden Unterkapitel erläutern die Grundlagen der verwendeten Programme.

### 4.7.1 Python

Python ist eine Programmiersprache, welche sich durch ihre klare Syntax und umfangreiche Bibliotheken auszeichnet und deshalb geeignet ist für wissenschaftliche Anwendungen. In der Astronomie wird Python für die Datenanalyse, die Modellierung und die Visualisierung verwendet. Durch die Kombination dieser Tools ist Python ein essenzielles Werkzeug für die Exoplanetenforschung und ist aus der modernen Astronomie nicht mehr wegzudenken.

### 4.7.2 Anaconda

Anaconda ist eine Python-Distribution, welche Pakete und Werkzeuge für wissenschaftliche Anwendungen enthält. Es vereinfacht die Paketverwaltung und erstellt isolierte Python-Umgebungen, sodass verschiedene Projekte getrennt bleiben können. Besonders praktisch ist die einfache Installation zahlreicher wissenschaftlicher Bibliotheken und die Verwaltung unterschiedlicher Python-Versionen innerhalb separater Umgebungen, was die Kompatibilität und Reproduzierbarkeit von Projekten gewährleistet (Anaconda Software Distribution, 2025).

### 4.7.3 JupyterLabs

JupyterLabs ist eine webbasierte Entwicklungsumgebung, welche die Programmierung und die Verbindung von Code, Visualisierungen und Text in Notebooks ermöglicht (Kluyver et al., 2016). Deshalb ist es ein ideales Tool für wissenschaftliche Untersuchungen und Datenanalysen. Es unterstützt eine Benutzeroberfläche, in welcher mehrere Programme und Anwendungen gleichzeitig ausgeführt werden können.

### 4.7.4 BATMAN

BATMAN (BAsic Transit Model cAlculation) ist ein Python-Paket zur Modellierung von Exoplanetentransits. Es berechnet Lichtkurven für beliebige Randverdunkelungs-Gesetze mithilfe eines Algorithmus, der auch für komplexe Randverdunkelungsprofile genaue Berechnungen erlaubt. Die Nutzer des Pakets können Transitparameter flexibel anpassen und so individuelle Transitmodelle erstellen. Das Paket bietet ausserdem Funktionen zur Anpassung der numerischen Genauigkeit, Supersampling für lange Belichtungszeiten sowie parallele Berechnung zur Beschleunigung grosser Modelle. BATMAN wird in der Exoplanetenforschung weit verbreitet genutzt, da es eine schnelle Simulation grosser Datenmengen ermöglicht und vielfältige Anwendungsfälle wie Sekundärfinsternisse unterstützt (Kreidberg, 2015). Die anwenderfreundliche API und die Open-Source-Verfügbarkeit tragen zur Etablierung als Standardwerkzeug für die Analyse von Transitlichtkurven bei.

### 4.7.5 Emcee

Emcee ist ein in Python implementierter Markov-Chain-Monte-Carlo-Sampler (MCMC), der häufig für probabilistische Modellierung und Parameterschätzung in der Astrophysik eingesetzt wird. Durch seinen affinen invarianten Ensemble-Sampling-Algorithmus benötigt EMCEE nur wenige Parameter und ist besonders effizient bei komplexen, hochdimensionalen Verteilungen. Das Paket erlaubt eine einfache Handhabung, parallele Ausführung und liefert genaue Posterior-Verteilungen zur Unsicherheitsbestimmung von Modellparametern. Emcee ist

ein Standardwerkzeug der Exoplanetenforschung, etwa bei der Auswertung von Transit- und Radialgeschwindigkeitsdaten (Foreman-Mackey et al., 2013).

#### **4.7.6 ExoFOP**

Das Exoplanet Follow-up Observing Program (ExoFOP) ist eine Datenbank der NASA zur Koordination von Nachbeobachtungen (NASA und California Institute of Technology, 2025). Forschende können dort Beobachtungsdaten und Analyseergebnisse austauschen, wodurch unnötige Messungen vermieden und Ressourcen effizient genutzt werden.

Die Plattform bietet Zugriff auf Datensätze zu TESS-Objekten, einschliesslich physikalischer Parameter und Beobachtungsnotizen (NASA und California Institute of Technology, 2025). Diese Struktur unterstützt die Validierung von Exoplanetenkandidaten durch unabhängige Nachweise. ExoFOP ist damit ein zentrales Werkzeug zur Bestätigung eines TOIs.

#### **4.7.7 Swarthmore College Transit Finder**

Der Swarthmore College Transit Finder ist ein Werkzeug, mit welchem die Berechnung beobachtbarer Transits von TOIs für einen gegebenen Beobachtungsstandort und Zeitraum erleichtert wird. Das Tool gibt den Transitstart, die Transitmitte, das Transitende, die Himmelskoordinaten, den Beobachtungszeitpunkt und den beobachtbaren Transitanteil an, sodass eine Beobachtung mit einem Bodenteleskop durchgeführt werden kann (Jensen, 2013).

Das Programm lässt ausserdem Nutzende auch individuelle Koordinaten und Parameter einstellen, sodass auch die Transitdaten eines einzelnen Objekts berechnet werden können. Das Tool ermöglicht Forschenden somit die optimalen Beobachtungsfenster für Exoplanetentransits zu berechnen, sodass die Effizienz der Datenerfassung gesteigert wird (Jensen, 2013).

#### **4.7.8 AstrolmageJ**

AstrolmageJ (AIJ) ist ein Open-Source-Programm zur astronomischen Bildverarbeitung mit Fokus auf die Durchführung von zeitreihenbasierter Differentialphotometrie (Collins et al., 2017). Das Programm ermöglicht die Verarbeitung von astronomischen Bilddaten, die Erzeugung von Lichtkurven und damit auch die Auswertung des beobachteten Transits. Ausserdem kann die Software auch mehrere Lichtquellen gleichzeitig analysieren sowie auch die aus den Bildern berechneten Werte trendbereinigen. Somit ist AIJ ein wichtiges Werkzeug für photometrische Analysen in der Exoplanetenforschung.

### **4.8 Lyceum 130 Observatorium, Nowosibirsk**

Bodengebundene Nachbeobachtungen sind ein unverzichtbarer Bestandteil der Exoplanetenforschung mit TESS. Der Grund liegt in der vergleichsweise groben Winkelauflösung des Satelliten: TESS besitzt eine Pixelskala von 21 Bogensekunden pro Pixel. Innerhalb einer typischen TESS-Photometrie-Apertur können sich daher mehrere Sterne befinden, die bei der Helligkeitsmessung zusammengefasst werden. Eklipsierende Doppelsterne (Eclipsing Binaries, EBs) oder Dreifachsternsysteme, die sich zufällig in der Nähe des Zielsterns befinden, können Transite imitieren, die auf den ersten Blick mit einem Exoplaneten verwechselt werden

können. Solche Falsch-Positive (engl. *Nearby Eclipsing Binaries*, NEBs) müssen durch hochauflösende Erdbeobachtungen identifiziert und ausgeschlossen werden (NASA und California Institute of Technology, 2025).

Das Lyceum 130 Observatorium (LHO) in Nowosibirsk, Russland, betrieben von Alexey Garmash, hat am 15. Oktober 2025 einen Transit von TOI-7265.01 im  $i_p$ -Band durchgeführt (Garmash, 2025). Das verwendete Teleskop hat einen Spiegeldurchmesser von 0,28 m. Die Beobachtungsapertur betrug 2,8 Bogensekunden (entsprechend 10 Pixeln), und die gemessene Halbwertsbreite (FWHM) lag bei 1,9 Bogensekunden (6,9 Pixel). Diese deutlich feinere Auflösung im Vergleich zu TESS ermöglicht es, benachbarte Störquellen räumlich aufzulösen und von der Zielquelle zu trennen.

Die Transitvorhersage basierte auf dem zu diesem Zeitpunkt verfügbaren Ephemeriden-Modell. Die Beobachtung in Nowosibirsk lieferte folgende Resultate:

**Tabelle 1:** Vergleich der vorhergesagten und gemessenen Transitparameter (LHO, 2025.10.15)

Parameter	Vorhergesagt	Gemessen	Abweichung
Transitmitte $T_c$ (BTJD)	10964,1151	10964,1212	+9 min (später)
Transittiefe (ppt)	46,0	46,5	+0,5 ppt
Transitdauer	1h 41min $\pm$ 7min	1h 54min	+13 min

Die gemessene Transitmitte lag rund 9 Minuten später als vorhergesagt. Dies liegt innerhalb der Messungenauigkeit und ist auf die noch nicht vollständig konvergierte Ephemeride zum Zeitpunkt der Vorhersage zurückzuführen. Die Transittiefe von 46,5 ppt stimmt gut mit dem vorhergesagten Wert von 46 ppt überein. Die gemessene Transitdauer von 1 Stunde und 54 Minuten ist etwas grösser als die Vorhersage von 1 Stunde und 41 Minuten, liegt jedoch innerhalb der Unsicherheiten des damaligen Modells.

Als Teil der Qualitätssicherung wurden alle Gaia-Sterne innerhalb von 2,50 Bogenminuten um TOI-7265 auf potenzielle NEB-Kontaminationen geprüft. Insgesamt wurden 271 Gaia-Sterne im Suchradius identifiziert. Davon konnten 126 Sterne eindeutig als nicht kontaminierend ausgeschlossen werden (*cleared*), 91 weitere Sterne wurden als wahrscheinlich unproblematisch eingestuft (*likely cleared*), und 48 Sterne konnten noch nicht endgültig ausgeschlossen werden (*not cleared*).

Eine besonders relevante Quelle ist ein Begleitstern (T2) im Abstand von 7,87 Bogensekunden vom Zielstern, mit einer TESS-Magnitude von  $T = 12,865$  und einer Helligkeitsdifferenz von  $\Delta T = -1,15$  mag gegenüber dem Zielstern. Dieser Stern liegt innerhalb der TESS-Pixelskala und könnte theoretisch zur gemessenen Lichtkurve beitragen. Die Bodenbeobachtung ermöglichte es jedoch, diesen Stern räumlich zu trennen und seinen Einfluss zu quantifizieren (Garmash, 2025).

Die Bildstabilität während der Beobachtungsnacht war gut: Die Standardabweichung des Bildmittelpunkts (Image Shift) betrug 0,6125 Bogensekunden in der X-Achse und 0,127 Bogensekunden in der Y-Achse. Diese geringen Verschiebungen belegen eine stabile Nachführung und minimale atmosphärische Bewegung, was die Qualität der Photometrie sicherstellt.

Alle Rohdaten, Lichtkurven und Analyseergebnisse dieser Beobachtung wurden auf der Plattform ExoFOP-TESS hochgeladen und stehen der wissenschaftlichen Gemeinschaft für weitere Analysen zur Verfügung (NASA und California Institute of Technology, 2025).

## 4.9 NuanceGP

Rohe TESS-Lichtkurven enthalten neben dem eigentlichen Transitsignal eine Vielzahl von systematischen Störungen. Diese stammen aus verschiedenen Quellen: instrumentelle Effekte des Satelliten (z. B. thermische Drift, Detektorartefakte), Schwingungen und Lageänderungen des Raumfahrzeugs (engl. *spacecraft jitter*), sowie intrinsische stellare Variabilität des Zielsterne selbst zum Beispiel Sternflecken oder Konvektion. Werden diese Trends nicht sorgfältig entfernt, können sie das Transitsignal verzerren, die gemessene Transittiefe verfälschen und die Parameterableitung beeinträchtigen.

Für diese Arbeit wurden TESS-Daten aus den Sektoren 55, 56, 75, 76 und 82 verwendet. Die Zeitangaben in den TESS-Daten liegen im Format BTJD (Barycentric TESS Julian Date), das mit dem üblichen BJD (Barycentric Julian Date) über folgende Umrechnung zusammenhängt:

$$\text{BJD} = \text{BTJD} + 2\,457\,000,0$$

Zur Bereinigung dieser Daten wurde das Python-Paket *Nuance* von Garcia et al. (2024) eingesetzt (Garcia et al., 2024). *Nuance* verwendet Gaußsche Prozesse (engl. *Gaussian Processes*, GP), um die korrelierte, langsam variierende Struktur in der Lichtkurve statistisch zu modellieren. Gaußsche Prozesse definieren eine Wahrscheinlichkeitsverteilung über Funktionen und sind besonders geeignet, glatte, nicht-lineare Trends zu beschreiben, ohne eine starre funktionale Form vorzugeben.

Das Kernkonzept besteht darin, dass der GP den “Hintergrund” der Lichtkurve modelliert. Das Transitsignal selbst wird dabei explizit im Modell eingebettet und während der GP-Anpassung geschützt, sodass es nicht versehentlich zusammen mit den Trends entfernt wird. Nach der Anpassung wird der GP-Trend von den Rohdaten abgezogen, und die resultierende bereinigte Lichtkurve enthält idealerweise nur noch das Transitsignal und weißes (unkorreliertes) Rauschen. Dieses Verfahren ist besonders robust für aktive Sterne und ist ein State-of-the-Art-Ansatz in der modernen Exoplanetenforschung (Garcia et al., 2024).

## 4.10 Reparametrisierungen für effizientes MCMC-Sampling

Beim MCMC-Sampling von Transitparametern treten systematische Schwierigkeiten auf, wenn der Parameterraum geometrisch ungünstig geformt ist. Stark gekrümmte oder “bananenförmige” Posteriores (engl. *banana degeneracy*) führen dazu, dass MCMC-Ketten langsam mischen, korrelierte Samples erzeugen und die Fehlerbalken unterschätzen. Durch geeignete Variablentransformationen lässt sich der Parameterraum so umformen, dass die Posteriores annähernd gauss’sch und dekorreliert werden. Dies verbessert Konvergenz, Effizienz und Zuverlässigkeit des Samplers erheblich.

### 4.10.1 Espinoza (2018): $(r_1, r_2) \rightarrow (b, p)$

Ein zentrales Problem beim gemeinsamen Sampling von Stossparameter  $b$  und Radiusverhältnis  $p = R_p/R_s$  ist die nicht-rechteckige Gestalt des physikalisch erlaubten Raums. Für einen Durchgangstransit muss gelten:

$$0 \leq b \leq 1 + p$$

Diese Bedingung erzeugt in der  $(b, p)$ -Ebene ein dreieckiges, schräg begrenztes Gebiet. Ein Sampler, der naiv in diesem Raum arbeitet, erzeugt stark korrelierte, bananenförmige Posteriores, die numerisch schwierig zu handhaben sind.

Espinoza (2018) schlägt eine bijektive Transformation vor, die zwei gleichverteilte Hilfsvariablen  $r_1, r_2 \in [0, 1]$  auf den physikalisch zulässigen  $(b, p)$ -Raum abbildet (Espinoza, 2018). Zunächst werden untere und obere Grenzen für das Radiusverhältnis definiert:  $p_L = 0,03$  und  $p_U = 0,28$ . Der Skalierungsparameter ist:

$$A_r = \frac{p_U - p_L}{2 + p_L + p_U}$$

Die Transformation erfolgt in zwei Fällen:

Fall 1 ( $r_1 > A_r$ ): Der Sampler befindet sich im oberen Teil des Parameterraums.

$$b = (1 + p_L) \left( 1 + \frac{r_1 - 1}{1 - A_r} \right), \quad p = (1 - r_2) p_L + r_2 p_U$$

Fall 2 ( $r_1 \leq A_r$ ): Der Sampler befindet sich im unteren Teil des Parameterraums, in dem kleine  $b$ -Werte und "streifende" Transitgeometrien möglich sind.

$$q = \frac{r_1}{A_r}, \quad b = (1 + p_L) \sqrt{q} r_2, \quad p = p_U + (p_L - p_U) \sqrt{q} (1 - r_2)$$

Da  $r_1$  und  $r_2$  gleichverteilt auf  $[0, 1]$  sind und die Jacobi-Determinante der Transformation konstant ist, werden alle physikalisch zulässigen  $(b, p)$ -Kombinationen gleichmässig abgedeckt. Das Resultat sind näherungsweise gauss'sche, dekorrelierte Posteriores in  $(r_1, r_2)$ , ohne bananenförmige Strukturen (Espinoza, 2018).

#### 4.10.2 Kipping (2013): $(q_1, q_2) \rightarrow (u_1, u_2)$

Das quadratische Limb-Darkening-Gesetz beschreibt die Helligkeitsabnahme eines Sterns vom Zentrum zum Rand als:

$$\frac{I(\mu)}{I(1)} = 1 - u_1(1 - \mu) - u_2(1 - \mu)^2$$

wobei  $\mu = \cos \theta$  der Kosinus des Winkels zwischen der Sichtlinie und der Sternoberfläche ist. Die physikalischen Constraints schränken den erlaubten Bereich der Koeffizienten  $(u_1, u_2)$  auf ein Dreieck im zweidimensionalen Parameterraum ein. Ein naiver Sampler, der beide Koeffizienten unabhängig und gleichverteilt zieht, würde häufig unphysikalische Werte vorschlagen und erzeugt korrelierte, bananenförmige Posteriores.

Kipping (2013) löst dieses Problem durch eine bijektive Transformation auf zwei neue Parameter  $q_1, q_2 \in [0, 1]$  (Kipping, 2013):

$$u_1 = 2\sqrt{q_1} q_2, \quad u_2 = \sqrt{q_1} (1 - 2q_2)$$

Die Umkehrtransformation lautet entsprechend:

$$q_1 = (u_1 + u_2)^2, \quad q_2 = \frac{u_1}{2(u_1 + u_2)}$$

Mit dieser Parametrisierung wird das Dreieck im  $(u_1, u_2)$ -Raum bijektiv auf das Einheitsquadrat in  $(q_1, q_2)$  abgebildet. Da der Sampler einfach  $q_1, q_2 \in [0, 1]$  zieht, werden automatisch nur physikalisch zulässige Limb-Darkening-Koeffizienten erzeugt, und der Prior ist gleichmässig im physikalisch erlaubten Raum. In der Praxis zeigen Corner-Plots der  $(q_1, q_2)$ -Posterioren annähernd gauss'sche, runde Konturen, während dieselben Ergebnisse in  $(u_1, u_2)$  stark bananenförmige Strukturen aufweisen (Kipping, 2013).

#### 4.10.3 Sozzetti et al. (2007): $\rho_\star \rightarrow a/R_\star$

Anstatt das Verhältnis  $a/R_\star$  (grosse Halbachse in Einheiten des Sternradius) direkt zu sampeln, verwenden wir die mittlere Sterndichte  $\rho_\star$  als Primärparameter und leiten  $a/R_\star$  daraus über das dritte Kepler'sche Gesetz ab (Sozzetti et al., 2007):

$$\frac{a}{R_\star} = \left( \frac{G\rho_\star P^2}{3\pi} \right)^{1/3}$$

Diese Reparametrisierung hat mehrere Vorteile. Erstens ist  $\rho_\star$  eine physikalisch interpretierbare Grösse, die direkt mit stellaren Modellen und spektroskopischen Messungen verglichen werden kann. Zweitens erlaubt ein einziger  $\rho_\star$ -Parameter, den Orbit gleichzeitig für alle Datensätze (verschiedene Instrumente, verschiedene Epochen) zu beschreiben, was die Zahl der freien Parameter im gemeinsamen Fit reduziert. Drittens ist die Posteriore in  $\rho_\star$  erfahrungsgemäss nahe gauss'sch und gut konditioniert (Sozzetti et al., 2007).

#### 4.10.4 Logarithmische Parameter

Weitere Reparametrisierungen betreffen Parameter mit positiver Definitionsbedingung und potenziell breitem Wertebereich. Für solche Grössen wird der natürliche Logarithmus als Samplingvariable verwendet:

- $\ln(\rho_\star)$ : Sterndichte; verhindert negative Samples, symmetrisiert unsichere Posterioren, und entspricht einem log-gleichverteilten Prior, der über mehrere Grössenordnungen informationsneutral ist.
- $\ln(P)$ : Periode; nützlich falls die Periode ist.
- $\ln(\sigma_w)$ : Jitter-Parameter (weisses Zusatzrauschen); verhindert negative Werte, erlaubt einen breiten Prior und erzeugt symmetrische Unsicherheiten im Log-Raum.

Durch diese logarithmischen Transformationen wird der Parameterraum gestreckt und symmetrisiert, sodass der MCMC-Sampler effizienter konvergiert und die Fehlerbalken korrekt widerspiegeln.

## 5 Methodik

Die Identifikation und Charakterisierung des Exoplanetkandidaten TOI 7265.01 erfolgte in mehreren Schritten, die von der Auswahl des Beobachtungsziels über die Datenerhebung bis zur statistischen Modellierung reichten. Die nachfolgenden Unterkapitel beschreiben diesen Prozess im Detail.

### 5.1 Auswahl des Exoplanetkandidaten

Die Auswahl eines möglichen Beobachtungsziels erfolgte mithilfe der ExoFOP-Datenbank. Für diese Arbeit wurden folgende Auswahlkriterien definiert:

- Sterntyp: M-Kategorie Sterne mit möglichst kleinem Radius und einer kühlen Temperatur
- Planetengröße: Möglichst grosse Exoplanetkandidaten (ungefähr jupitergross)
- Beobachtungsstatus: Es sollte keine photometrischen Beobachtungen des Exoplanetkandidaten geben, sodass die Arbeit die erste Bestätigung darstellt.

Diese Konstellation ist wissenschaftlich interessant, da es bisher kein zufriedenstellendes physikalisches Modell für Riesenplaneten um kleine, kühle Sterne gibt (Schlecker et al., 2022). Um die ExoFOP-Datenbank nach passenden Exoplanetkandidaten zu filtern, wurde diese als Excel-Datei heruntergeladen, und nach den oben aufgelisteten Kriterien gefiltert. Nach der Filterung gab es insgesamt zehn Exoplanet-Stern-Paare, welche allen Kriterien genau entsprechen.

### 5.2 Beobachtungsplanung

Im Rahmen der Beobachtungsplanung musste aus der Liste der zehn Exoplanetkandidaten derjenige ausgewählt werden, der sich am besten für eine Beobachtung mit dem Saint-Ex-Observatorium eignet. Diese Entscheidung basiert auf verschiedenen Kriterien, wie der Transitdauer, den Beobachtungsfenstern und der Sichtbarkeit des Standortes. Der am besten geeignete Kandidat wurde schliesslich observiert und analysiert.


#### 5.2.1 Berechnung der Transitzeiten und Bestimmung des Beobachtungsziels

Die Planung der Beobachtung erforderte die Bestimmung der Zeitpunkte, in denen die zehn Exoplanetkandidaten vom Saint-Ex-Observatorium beobachtbar sind. Weil das Saint-Ex-Observatorium ein etabliertes Forschungsteleskop ist, gibt es im Swarthmore College Transit Finder bereits vorgegebene Werte für die Höhe und geographischen Koordinaten dafür. Sonst müsste man diese Werte manuell in den Transit Finder eingeben. Nur so konnte der bestgeeignete Kandidat bestimmt werden und dem Forschungsteam in Mexiko weitergeleitet werden.

Um den bestgeeigneten Exoplanetkandidat zu bestimmen, sollte der gesamte Transit in einer Beobachtung des Saint-Ex-Observatoriums beobachtbar sein. Um die Transitzeiten zu berechnen, wurde der Transit Finder des Swarthmore Colleges verwendet, da dieser ohne grosse Mühe mögliche Observationsdaten berechnen kann. Die jeweilig benötigten Daten zur Berechnung der Transitzeiten wurden aus ExoFOP entnommen und in die für den Swarthmore College Transit Finder passenden Einheiten umgewandelt.

Danach wurden nur für fünf der ursprünglichen zehn TOIs Transitzeiten gefunden, wobei nur bei zwei TOIs der gesamte Transit in einer Observation beobachtbar war. Diese Exoplanetkandidaten waren TOI 7316.01 und TOI 7265.01. TOI 7316.01 schien nach Absprache mit der externen Betreuung aufgrund seiner Beobachtungsumstände der schlechtere der beiden Kandidaten zu sein. Deshalb hat sich TOI 7265.01 als Beobachtungsziel durchgesetzt.

**Abbildung 7:** Output des Swarthmore College Transit Finders für TOI 7265.01.

Local evening date	Name	V or Gaia mag	Start—Mid—End	Duration	BJD <sub>TDB</sub> start—mid—end	Elev. at start, mid, end ±1 hrs	% of transit (baseline) observable. Suggested obs. start, end	Az. at start, mid, end ±1 hrs	HA at start, mid, end ±1 hrs	RA & Dec (J2000)	Period (days)	Depth (ppt)	Comments
Wed. 2025-07-02: Nautical twilight 2025-07-02 20:51 — 2025-07-03 04:41 local time / 2025-07-03 03:51 — 2025-07-03 11:41 UTC													
Wed. 2025-07-02 Nautical twilight 20:51 – 04:41 (America/Tijuana)	<a href="#">TOI 7265.01</a> Finding charts: <a href="#">Annotated</a> , <a href="#">Aladin</a> , <a href="#">SkyMap</a> , <a href="#">Airmass plot</a> , <a href="#">ACP plan</a> Info: <a href="#">Exoplanet Archive</a> , <a href="#">Simbad</a> , <a href="#">Gaia</a> , <a href="#">TIC</a>	Moon 54% @120°	22:29 23:29— 00:20 —01:10 02:10	1:41	10859.7723 10859.8074 10859.8425	35° 45° 54° 62° 71°	 100% (100%) 22:29—02:10	53° 54° 53° 49° 34°	-4.7 -3.7 -2.9 -2.0 -1.0	21:14:48.920 +45:55:33.72	4.17	46257000000	Manually-entered single object

In Abbildung 7 kann man die relevanten Daten für die Observation ablesen. Diese wären die Beobachtungszeit, die Transittlänge, die Himmelsposition von TOI 7265.01, der beobachtbare Anteil des Transits und die Transittiefe.

### 5.3 Datenerhebung

Die Beobachtung wurde in der Nacht des 2. Juli 2025 von der Forschungsgruppe des Saint-Ex-Observatoriums durchgeführt. Dabei wurden insgesamt 534 Einzelaufnahmen des Himmelsausschnitts um TOI 7265.01 erstellt. Jede Aufnahme zeigt den Zielstern sowie mehrere Referenzsterne, die später für die Differenzialphotometrie verwendet werden.

**Abbildung 8:** Eine Rohaufnahme des Sternfelds um TOI 7265.01.



Bei der Durchsicht der Rohdaten wurde jedoch festgestellt, dass ein Bild durch einen hellen Lichtstreifen verunreinigt war. Diese Verunreinigung wurde vermutlich durch einen Satelliten oder durch einen Meteor verursacht.

**Abbildung 9:** Verunreinigtes Rohbild mit heller Störspur, vermutlich durch einen Satelliten- oder Meteoritenflug verursacht.



In Abbildung 9 erkennt man deutlich, wie sich im Bild ein langer und heller Streifen durchsetzt. Aufgrund dessen wurde dieses spezifische Bild nicht für die weitere Analyse verwendet.

## 5.4 Datenreduktion und Bildkalibrierung

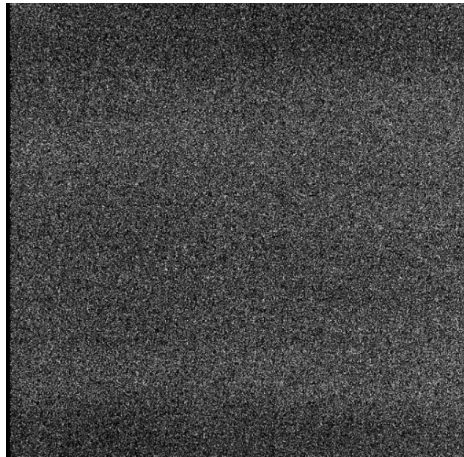
Im Anschluss an die Datenerhebung wurden die Rohbilder kalibriert und von systematischen Messfehlern befreit. Die Datenreduktion ist ein essenzieller Schritt, um später eine photometrische Analyse durchführen zu können. Dazu gehört die Korrektur mit Bias-, Dark- und Flat-Bildern, welche typische Kameraartefakte, Störungen und eine ungleichmässige Sensitivität der Detektoren entfernen.

### 5.4.1 Dark-Frames

Ein Dark-Frame ist eine Aufnahme mit geschlossenem Verschluss, bei der kein Licht auf den Kamerasensor fällt. Somit wird nur das thermische Rauschen der Kamera aufgenommen, welches mit der Belichtungszeit zunimmt (Practical Astrophotography Staff, 2015). Dieses Rauschen entsteht durch thermische Elektronen, die auch ohne Lichtsignal im Sensor entstehen.

Dark-Frames werden mit derselben Belichtungszeit und Temperatur aufgenommen wie die Rohbilder (Practical Astrophotography Staff, 2015). Mehrere Dark-Frames werden von AIJ zu einem Master-Dark-Frame kombiniert. Von AIJ wird dann das Master-Dark-Frame von den wissenschaftlichen Bildern subtrahiert, um das thermische Rauschen zu entfernen.

**Abbildung 10:** *Das Master-Dark-Frame für die Beobachtung von TOI 7265.01.*

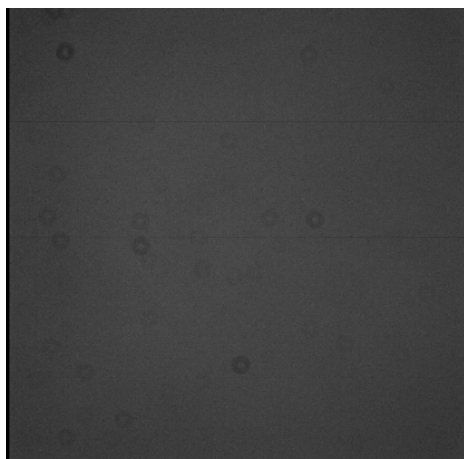


In der Beobachtung von TOI 7265.01 wurden insgesamt 79 Dark-Frames erstellt. Diese wurden von AIJ zu einem Master-Dark-Frame zusammengestellt, welches in Abbildung 10 erkennbar ist.

#### **5.4.2 Flat-Frames**

Ein Flat-Frame ist eine Aufnahme einer gleichmässig ausgeleuchteten Fläche, wie etwa des Dämmerungshimmels oder einer beleuchteten Innenkuppel. Es zeigt, wie die verschiedenen Empfindlichkeiten der einzelnen Pixel des Kamerasensors sowie Vignettierung und Staubpartikel auf der Optik einen Fehler erzeugen (Practical Astrophotography Staff, 2015). Durch die Zusammenfassung durch AIJ aller Flat-Frames aus einer Beobachtung kann man ein Master-Flat-Frame erstellen. Dieses wird dann von den wissenschaftlichen Rohdaten subtrahiert.

**Abbildung 11:** *Das Master-Flat-Frame für die Beobachtung von TOI 7265.01.*



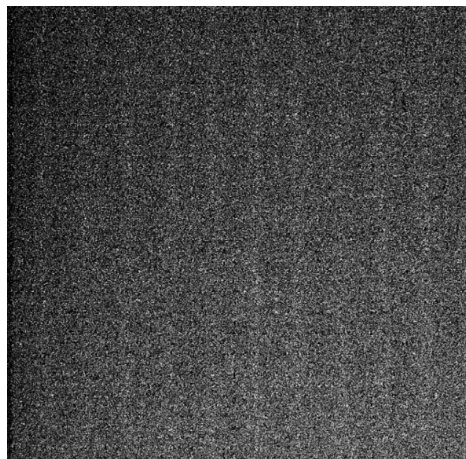
Für die Beobachtung von TOI 7265.01 wurden insgesamt 37 Flat-Frames aufgenommen. In Abbildung 11 ist der Master-Flat-Frame für die Observation von TOI 7265.01 zu erkennen. Es gibt einen deutlichen Unterschied zwischen den Dark- und Bias-Frames, da hier nicht ein Rauschen aufgenommen wird, sondern andere Störfaktoren.

### 5.4.3 Bias-Frames

Ein Bias-Frame ist ein Bild, welches mit minimaler Belichtungszeit aufgenommen wird. Dadurch wird das elektronische Rauschen der Kamera, welches bei jeder anderen Aufnahme vorhanden ist (Practical Astrophotography Staff, 2015). Um dieses Rauschen herauszufiltern, muss man mit AIJ diese Bias-Frames von den Flat-Frames subtrahieren.

Üblicherweise wird ein Master-Bias-Frame kreiert, welcher den Mittelwert aller generierten Bias-Frames darstellt. Der Master-Bias wurde anschliessend von allen Flat-Frames subtrahiert, um das elektronische Grundrauschen zu entfernen.

**Abbildung 12:** *Das Master-Bias-Frame für die Beobachtung von TOI 7265.01.*



Für die Beobachtung von TOI 7265.01 wurden insgesamt zehn Bias-Frames generiert. In Abbildung 12 erkennt man den Master-Bias-Frame, mit welchem jedes Flat-Frame korrigiert wurde.

### 5.4.4 Kalibrierte Bilder

Nach Anwendung der Bias-, Dark- und Flat-Frame-Korrekturen lagen kalibrierte Bilder vor, welche frei von systematischen Instrumentenfehlern waren. Diese 533 Bilder stellen die Grundlage für die photometrische Analyse dar.

**Abbildung 13:**

*Das Sternfeld um TOI 7265.01 nach Dark-, Bias- und Flat-Kalibrierung.*



Die Korrektur durch Bias-, Dark- und Flat-Frames wurde durch AIJ durchgeführt, welche eine simple Pipeline für die Bildreduktion astronomischer Aufnahmen bietet.

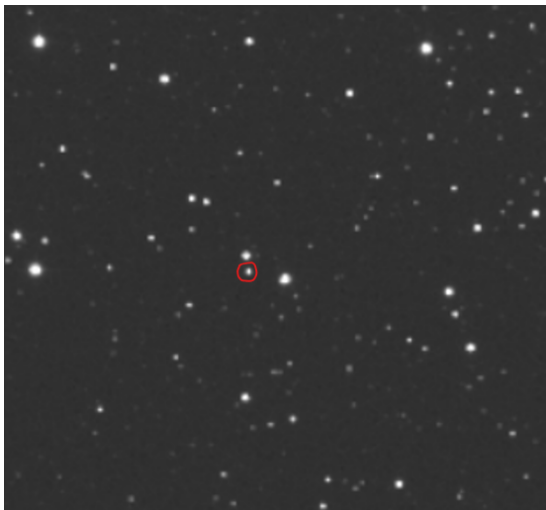
## 5.5 Photometrische Analyse

In diesem Kapitel wird die photometrische Analyse des Zielsterns TOI 7265.01 beschrieben. Diese wurde mit dem «Differential Photometry Tool» in AIJ durchgeführt. Dabei erfolgte zunächst die präzise Identifikation des Zielsterns in den bereinigten Bildern. Anschliessend wurde mithilfe der Differenzialphotometrie die relative Helligkeit über den Beobachtungszeitraum genau gemessen, um Transitparameter und andere Eigenschaften des Systems zu bestimmen. Die Methode ermöglicht eine zuverlässige Analyse der Lichtkurven und bildet die Grundlage für die weiterführende Auswertung der exoplanetarischen Daten. Das «The AstrolmageJ Cookbook» (Boyce & Conti, 2016) wurde als Guide verwendet, um AIJ zu verwenden. Die einzelnen Schritte dazu werden in den folgenden Unterkapiteln erläutert.

### 5.5.1 Identifikation des Zielsterns im Bild

Die Identifikation des Zielsterns in den bereinigten Bildern erfolgte mithilfe des im Swarthmore College Transit Finder integrierten «Aladin sky atlas». Aladin ist ein interaktiver Sternenatlas, in welchem man digitale Sternbilder, astronomische Kataloge und Datenbanken einsehen kann (Bonnarel et al., 2000). Durch Aladin wurde auf Sternbilder zugegriffen, in welchen der Zielstern markiert war. Dadurch wurde ermöglicht, auf den bereinigten Bildern den Zielstern zu identifizieren. Die Schwierigkeit liegt darin, dass die Himmelsrichtungen auf den bereinigten Bildern nicht dieselben sein müssen wie die Himmelsrichtungen auf den digitalen Sternbildern von Aladin.

**Abbildung 14:** Die um  $90^\circ$  gedrehte Eigenbeobachtung mit rotmarkiertem Zielstern.



**Abbildung 15:** Die aus Aladin stammende Identifikationshilfe.



Wie man in den Abbildungen 14 & 15 erkennt, ist das vom Saint-Ex-Observatorium aufgenommene Foto um circa  $90^\circ$  verdreht. Nun kann mit einem geübten Auge exakt dieselbe Sternkonstellation im digitalen Bild von Aladin und in dem um  $90^\circ$  gedrehten Bild aus der Eigenbe-

obachtung erkennen. So konnte TOI 7265.01 im Bild für die Differenzialphotometrie markiert werden.

### 5.5.2 Differenzialphotometrische Analyse mit AIJ

Die Helligkeitsmessung erfolgt mittels Differenzialphotometrie. Bei dieser Methode wird die absolute Helligkeit des Zielstern gemessen, sondern relativ zu mehreren Referenzsternen im selben Bild ermittelt.

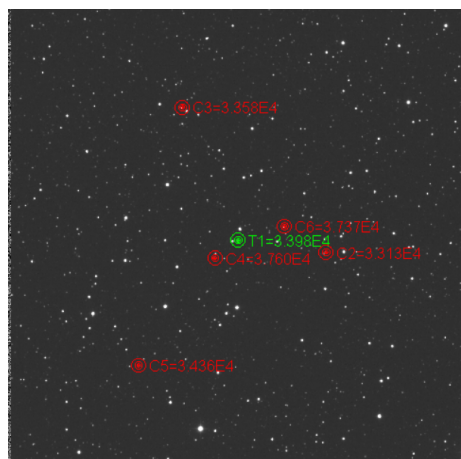
AIJ bietet ein Multi-Aperture-Photometry-Tool, welches die gleichzeitige Messung mehrerer Sterne im selben Bildfeld ermöglicht. Für jeden Stern wird die Gesamthelligkeit innerhalb eines kreisförmigen Bereiches (Apertur). Die relative Helligkeit des Zielsterns ergibt sich aus einer Division durch die kombinierte Helligkeit der Referenzsterne:

$$F_{\text{relativ}}(t) = \frac{F_{\text{Ziel}}(t)}{\sum F_{\text{Referenz}}(t)}$$

Diese Methode liefert eine Lichtkurve, welche die Helligkeitsänderung des Zielsterns im Vergleich zu der verstrichenen Zeit darstellt. Jedoch erfolgt die Analyse nicht automatisch durch AIJ. Zuerst muss das Multi-Aperture-Photometry-Tool angeklickt werden und dann auf den bereits identifizierten Zielstern geklickt werden. Nach dem Drücken der Entertaste erscheint ein Bild, in welchem AIJ die Aperturen für den Zielstern und fünf weitere Referenzsterne einzeichnet.

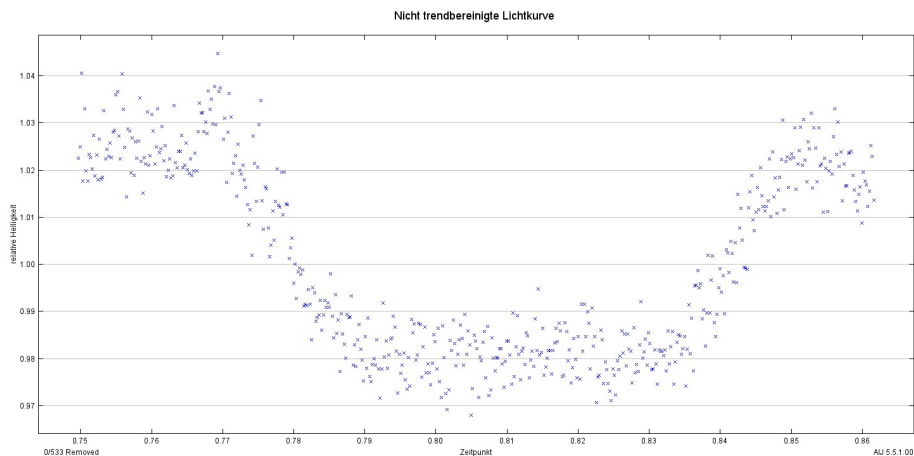
#### Abbildung 16:

*Die eingezeichneten Aperturen des Zielsterns (T1) und der Referenzsterne (C#).*



Nach erneutem Drücken der Entertaste beginnt AIJ damit, die relative Helligkeit des Zielsterns im Verlauf der Zeit als Diagramm zu zeichnen. Als Resultat entsteht eine nicht trendbereinigte Lichtkurve, bei welcher die Messdaten noch nicht nach weiteren systematischen Einflüssen gefiltert wurden.

**Abbildung 17:**  
*Unbereinigte Lichtkurve des Zielsterns während des Transits.*

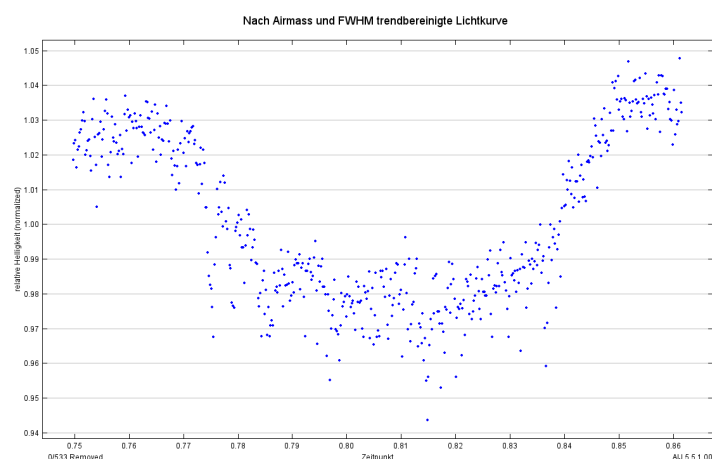


Die Lichtkurve aus Abbildung 17 enthält noch Schwankungen, welche durch physikalische Störeffekte verursacht werden. Um diese Störungen zu reduzieren, erfolgte eine Trendbereinigung in AIJ, bei der mit Hilfe von zwei Parametern störende Korrelationen aus der Lichtkurve entfernt wurden: der Airmass-Parameter und der Full Width at Half Maximum (FWHM).

Airmass-Parameter (Luftmasse) beschreibt, wie viel Atmosphäre das vom Stern emittierte Licht vor dem Eintreffen am Teleskop durchdringt. Eine höhere Luftmasse bedeutet mehr durchdrungene Luft und somit eine stärkere Lichtstreuung, sodass der Fehler grösser ist. Je höher das beobachtete Objekt am Himmel ist, desto kleiner ist dieser Fehler, da im Zenit die kleinste Luftmasse passiert werden muss.

Der FWHM gibt die Breite der Sternabbildung an und ist ein Mass für die Bildschärfe. Schwankungen der Bildschärfe verursachen ebenfalls Fehler in den Messwerten.

**Abbildung 18:**  
*Durch Luftmasse und FWHM trendbereinigte Lichtkurve.*



Nach der Trendbereinigung durch das in AIJ integrierte Tool wurden die beiden Störungsfaktoren aus den Messwerten entfernt, wodurch eine glattere und realitätsnähere Lichtkurve entstanden ist.

Auf Basis der Daten in den Abbildungen 17 & 18 wurden Simulationen des Transits in Python

durchgeführt, welche die Parameter des Exoplaneten berechnen, welche durch die Transitmethode kalkulierbar sind. Diese Simulation wird im folgenden Kapitel erläutert.

## 5.6 Simulation des Transits in Python

In diesem Kapitel wird die Simulation des Transits von TOI 7265.01 beschrieben. Mit dem selbstverfassten Python-Code wird eine Lichtkurve erzeugt, welche den Helligkeitsabfall während des Transits abbildet. Dabei kommen Methoden der Photometrie und ein MCMC-Verfahren zur Parameterschätzung zum Einsatz. Der Code ermöglicht eine numerische und grafische Auswertung der Transitparameter.

### 5.6.1 Importieren und Einlesen der Daten

Der erste Schritt besteht im Importieren der Python-Bibliotheken und im Einlesen der Messdaten, welche vom Saint-Ex-Observatorium aufgenommen wurden. Der Code liest zunächst eine Datei im Tab- oder Leerzeichen-getrennten Format ein, wobei Kommentarzeilen ignoriert werden. Die Spaltennamen werden bereinigt, um Leerzeichen zu entfernen, da diese in AIJ-Daten oft vorkommen.

#### Listing 1: Code zur Datenvorbereitung

```
# Importieren der Bibliotheken:
import batman
import emcee
import astropy
import scipy.stats as stats
import numpy as np
import lightkurve as lk
import matplotlib.pyplot as plt
import pandas as pd
try:
    import corner
except ImportError:
    raise SystemExit("pip install corner")

# Datenspeicherort angeben:
filename="/home/nikkn/sim/TOI_7265_01.tbl"

# Definieren einer Funktion, welche die AIJ-Tabelle in eine Lichtkurve
umwandelt:
def ai_j_tbl_to_lightcurve(filename, target_suffix="T1", normalize=True):
    df = pd.read_csv(filename, sep=r'[\t]+|_{2,}', engine='python',
        comment='#')
    df.columns = [c.strip().replace('_', '_')] for c in df.columns]

    # Liest die Zeitspalte und rechnet diese in BTJD um:
    # Bevorzugt BJD_TDB, falls vorhanden, ansonsten wird das Juliandatum
    # gebraucht.
    if 'BJD_TDB' in df.columns:
        t = df['BJD_TDB'].to_numpy(float) - 2457000.0
    elif 'J.D.-2400000' in df.columns:
        t = df['J.D.-2400000'].to_numpy(float) + 2400000.0 - 2457000.0
    elif 'JD.UTC' in df.columns:
        t = df['JD.UTC'].to_numpy(float) - 2457000.0
```

```

elif 'JD' in df.columns:
    t = df['JD'].to_numpy(float) - 2457000.0
else:
    raise ValueError("No recognizable time column (looked for BJD_TDB)")

fcol = f"rel_flux_{target_suffix}"
fecol = f"rel_flux_err_{target_suffix}"
if fcol not in df.columns:
    candidates = [c for c in df.columns if c.lower() == fcol.lower()]

if not candidates:
    raise ValueError(f"Could not find flux column '{fcol}' in columns")
fcol = candidates[0]

if fecol not in df.columns:
    candidates = [c for c in df.columns if c.lower() == fecol.lower()]
    fecol = candidates[0] if candidates else None
f = df[fcol].to_numpy(float)
fe = df[fecol].to_numpy(float) if fecol else None

# Erstellen einer Lichtkurven-Tabelle
lc = lk.LightCurve(time=t, flux=f, flux_err=fe).remove_nans()
if normalize:
    lc = lc.normalize()

return lc, df

```

Es wird geprüft, ob eine Zeitspalte existiert, welche die Zeit in Julian-Tageszahlen enthält und diese in das Baryzentrische Juliandatum (BTJD) umrechnet. So wird eine einheitliche Zeitskala erstellt. Falls keine Zeitspalte existiert, wird ein Fehler ausgelöst.

## 5.6.2 Festlegen der Startwerte für das Transitmodell

Im Rahmen der Simulation werden zu Beginn wesentliche Modellparameter initialisiert. Diese primitive Schätzungen der Planetenparameter wurden von ExoFOP übernommen, sodass die Simulation schneller zu einem vernünftigen Ergebnis kommt. Die Periode des Umlaufs, `period_guess`, wird auf 4.1723 Tage gesetzt und stellt eine erste Schätzung aufgrund der Lichtkurvenanalyse dar. Mit `t0_guess` erfolgt die Definition des Transitmittelpunkts, ausgehend vom beobachteten Transitzeitpunkt und korrigiert um den Offset zum BTJD-Nullpunkt. Die Tiefe des Transits, `depth_guess`, kodiert den relativen Flussabfall während eines Transits und liefert damit einen direkten Hinweis auf das Flächenverhältnis zwischen Planet und Stern. Weiter wird mit `log_dur_guess` der natürliche Logarithmus der Transiddauer in Tagen bestimmt, da die logarithmische Darstellung numerisch stabilere Fits erlaubt.

Die Variablen `x`, `y` und `yerr` extrahieren aus der Lichtkurve die Zeitpunkte, normalisierten Helligkeit und deren Unsicherheiten. Anschliessend werden mit `x_min`, `x_max` und `x_mid` die Ränder und die Mitte des Beobachtungszeitraums berechnet; `t0_ref` gibt den temporalen Bezugstransit an, indem der Mittelpunktzeitpunkt der Daten auf den nächsten theoretischen Transit projiziert wird. Für die physikalische Konsistenz werden zudem die Gravitationskonstante  $G$  und die Sonnendichte `RHO_SUN` als Referenzgrößen festgelegt.

## Listing 2: Initialisierung der Modellparameter

```
period_guess = 4.1723
t0_guess = 2460576.091201 - 2457000
depth_guess = 0.046257
log_dur_guess = np.log(1.684/24)
x = lc.time.value
y = lc.flux.value
yerr = lc.flux_err.value
x_min = np.min(x)
x_max = np.max(x)
x_mid = 0.5 * (x_min + x_max)
t0_ref = t0_guess + period_guess * np.round((x_mid - t0_guess) /
    period_guess)
G = 6.67430e-11
RHO_SUN = 1408
```

Diese Parameter garantieren, dass das Transitmodell auf einer relevanten ersten Näherung aufbaut und ermöglichen anschliessend ein robustes numerisches Fitten an die Beobachtungsdaten.

### 5.6.3 Transitsimulation und Modellimplementierung

Die Funktion `_safe_float_array(x, like=None)` wandelt eine Eingabe in ein Array vom Typ `float` um und prüft optional die Formkongruenz zu einer Referenzvariable `like`. Dadurch wird sichergestellt, dass die Datenstrukturen für die Transitmodellierung kompatibel sind. Die Funktion `transit_model(theta, x)` berechnet die Lichtkurve für ein gegebenes Parameterset `theta` und Beobachtungszeitpunkte `x`.

Die Parameter in `theta` umfassen `offset` für additive Basisverschiebung, `rho_sol` als normierte Sternendichte, `u1` und `u2` als Limbusabdunklungskoeffizienten, `logP` für logarithmische Umlaufperiode, `t0` für den Transitmittelpunkt, `logT14` für logarithmische Tr

ansitdauer, `b_sq` für das Quadrat des Impact-Parameters und `logdepth` für die logarithmische Transittiefe.

Die Funktion berechnet zunächst die Periode (`Pdays`) und deren Sekundenversion `Psec`. Der Impact Parameter `b` und der Winkel `mu` zum Mittelpunkt werden bestimmt. Anhand der Limbusabdunklungskonstanten wird die lokale Intensität auf der Sternscheibe (`ld`) berechnet. Sind die Werte unphysikalisch, wird eine Array mit NaN-Werten zurückgegeben.

## Listing 3: Kombinierte Transitmodellfunktion

```
def _safe_float_array(x, like=None):
    a = np.asarray(x, dtype=float)
    if like is not None and a.shape != np.shape(like):
        raise ValueError("shape mismatch")
    return a

def transit_model(theta, x):
    offset, rho_sol, u1, u2, logP, t0, logT14, b_sq, logdepth = theta
    [:9]
    x = _safe_float_array(x)
    Pdays = float(np.exp(logP))
    Psec = Pdays * 86400.0
    b = float(np.sqrt(max(b_sq, 0.0)))
    mu = np.sqrt(max(0.0, 1.0 - b**2))
```

```

ld = 1.0 - u1 * (1.0 - mu) - u2 * (1.0 - mu)**2
if not np.isfinite(ld) or ld <= 0:
    return np.full_like(x, np.nan, dtype=float)

depth = float(np.exp(logdepth))
ror = np.sqrt(depth / ld)

rho = float(rho_sol) * RHO_SUN
val = (G * rho * Psec**2) / (3.0 * np.pi)
if (not np.isfinite(val)) or (val <= 0):
    return np.full_like(x, np.nan, dtype=float)

aR = np.cbrt(val)
if (b < 0) or (b > 1.0 + ror) or (b > aR) or (not np.isfinite(aR)):
    return np.full_like(x, np.nan, dtype=float)

cosi = np.clip(b / aR, -1.0, 1.0)
inc_deg = float(np.degrees(np.arccos(cosi)))

if not (0.0 < u1 < 1.0 and (u1 + u2) < 1.0 and (u1 + 2.0*u2) > 0.0):
    return np.full_like(x, np.nan, dtype=float)

p = batman.TransitParams()
p.t0 = float(t0)
p.per = Pdays
p.rp = float(ror)
p.a = float(aR)
p.inc = inc_deg
p.ecc = 0.0
p.w = 90.0
p.u = [float(u1), float(u2)]
p.limb_dark = "quadratic"

m = batman.TransitModel(p, x)
lc = m.light_curve(p).astype(float, copy=False)

arg = np.clip((1.0/aR) * np.sqrt(max(0.0, (1.0 + ror)**2 - b**2)),
              -1.0, 1.0)
T14_days = (Pdays/np.pi) * np.arcsin(arg)

return float(offset) + lc

```

Die Transittiefe wird aus `logdepth` exponiert und in `depth` gespeichert. Daraus folgt `ror`, das Verhältnis von Planeten- zu Sternradius, unter Berücksichtigung der lokalen Intensität `ld`.

Die Sternendichte wird normiert und daraus mit Keplers drittem Gesetz das Verhältnis Bahnradius zu Sternradius  $a/R_*$  berechnet. Nach Plausibilitätsprüfungen wird die Bahnneigung (`inc_deg`) berechnet. Die Gültigkeit der Limb-Darkening-Koeffizienten wird geprüft.

Anschliessend wird das Transitmodell mit `batman` gebildet und die Lichtkurve erzeugt. Die Transitdauer  $T_{14}$  wird optional berechnet und zurückgegeben wird die Lichtkurve inklusive Offset.

#### 5.6.4 Log-Wahrscheinlichkeit und Modellvalidierung

Die Funktion `log_probability(theta, x, y, yerr)` bestimmt die logarithmische Wahrscheinlichkeit eines Parametersatzes `theta` anhand der beobachteten Daten `x, y` und deren Unsicher-

heiten `yerr`. Dazu werden zunächst alle Eingaben mit `_safe_float_array` in Fließkommazahlen umgewandelt und die Unsicherheiten beschränkt, um numerische Probleme zu vermeiden.

Im Anschluss werden die Priors für die einzelnen Parameter definiert. Die additive Offsetgröße `offset` wird als Normalverteilung mit Mittelwert 0 und Standardabweichung 0.01 angenommen. Die Sternendichte `rho_star` erhält eine Prior mit Mittelwert 5.81 und Standardabweichung 2.0. Die Randverdunkelungs-Koeffizienten `u1` und `u2` werden auf physikalisch zulässige Werte eingeschränkt und mit Normalverteilungen mit Mittelwerten 0.27 bzw. 0.18 und Standardabweichungen von 0.15 besetzt. Die logarithmische Perioden- und Transitmitte sowie die logarithmische Transiddauer erhalten Priors mit Mittelwerten aus Schätzungen und angemessenen Standardabweichungen. Nicht zulässige Wertebereiche für `b_sq` und `logdepth` führen zu einer sofortigen Ablehnung des Parametersets.

Das Modell wird mit `transit_model` generiert. Führt dieses zu nicht endlichen Werten der Lichtkurve, wird das Parameter-Set verworfen. Die Residuen zwischen Beobachtungs- und Modellwerten werden gewichtet und zur Berechnung der Log-Likelihood verwendet. Das endgültige Logarithmus der Posterior-Wahrscheinlichkeit ergibt sich als Summe von Prior und Likelihood.

#### Listing 4: Logarithmische Wahrscheinlichkeitsfunktion für MCMC

```
def log_probability(theta, x, y, yerr):
    x = _safe_float_array(x)
    y = _safe_float_array(y, like=x)
    yerr = np.clip(_safe_float_array(yerr, like=x), 1e-12, np.inf)
    offset, rho_star, u1, u2, logP, t0, logT14, b_sq, logdepth = theta

    # Priors
    lp = stats.norm.logpdf(offset, 0.0, 0.01)
    lp += stats.norm.logpdf(rho_star, 5.81, 2.0)
    if not 0.0 < u1 < 1.0 or (u1 + u2) >= 1.0 or (u1 + 2*u2) <= 0.0:
        return -np.inf
    lp += stats.norm.logpdf(u1, 0.27, 0.15)
    lp += stats.norm.logpdf(u2, 0.18, 0.15)
    lp += stats.norm.logpdf(logP, np.log(period_guess), 1e-1)
    lp += stats.norm.logpdf(t0, t0_ref, 2e-1)
    lp += stats.norm.logpdf(logT14, log_dur_guess, 0.2)
    if not (0.0 <= b_sq <= 1.0) or not (-5.0 <= logdepth <= -0.5):
        return -np.inf

    # Modellberechnung
    lc_model = transit_model(theta, x)
    if not np.all(np.isfinite(lc_model)):
        return -np.inf

    resid = (y - lc_model) / yerr
    loglike = -0.5 * np.sum(resid**2 + np.log(2*np.pi*yerr**2))
    return lp + loglike
```

Diese Funktion stellt die Bewertungsgrundlage für MCMC-Verfahren dar.

### 5.6.5 MCMC-Sampling zur Parameterschätzung

Für die Parameterbestimmung wird der `emcee`-Sampler verwendet. Dabei werden neun Dimensionen (`ndim = 9`) mit 32 sogenannten Walkern (`nwalkers = 32`) und insgesamt 50000 Schritten (`nsteps = 50000`) definiert. Ein Burn-in von 10000 Schritten eliminiert die Anfangsphase

des Samplings, um die Wanderer in den stationären Bereich zu bringen.

Die Startwerte `initial_pos` des Samplers beinhalten aufeinander abgestimmte geschätzte Parameter, wie den additiven Offset, die normierte Sternendichte `rho_star` in Sonnen-Dichten, die Limb-Darkening-Koeffizienten `u1` und `u2`, logarithmische Werte der Periode (`logP`), des Transitmittelpunktes (`t0`), der Transiddauer (`logT14`), den quadrierten Stossparameter (`b_sq`) und die logarithmische Transittiefe (`logdepth`). Kleinere Zufallsschritte sorgen für die Verteilung der Walker in der Parameterwolke.

Mit dem `EnsembleSampler` wird die Log-Wahrscheinlichkeitsfunktion `log_probability` als Bewertungsmetrik verwendet, wobei die Daten `x` übergeben werden. Nach Abschluss der Samplingphase werden die Samples extrahiert, wobei die Burn-in-Phase verworfen und die Ketten geglättet werden.

Aus den Samples der logarithmischen Transittiefe wird die Verteilung des Verhältnisses von Planeten- und Sternradius `ror_samples` durch Exponentiation berechnet, da diese als halber Logarithmus im Sampling vorliegt.

#### Listing 5: MCMC-Parameter-Sampling mit `emcee`

```
ndim = 9
nwalkers = 32
nsteps = 50000
burnin = 10000

initial_pos = np.array([
    0.0, # offset (additive)
    5.18, # rho_star [rho_sun]
    0.27, 0.18, # u1, u2
    np.log(period_guess), # logP
    t0_ref, # t0
    log_dur_guess, # logT14
    0.3**2, # b_sq
    np.log(0.0468) # log_depth
], dtype=float)

pos = initial_pos + 1e-4 * np.random.randn(nwalkers, ndim)
sampler = emcee.EnsembleSampler(nwalkers, ndim, log_probability, args=(x, y, yerr))
sampler.run_mcmc(pos, nsteps, progress=True)
samples = sampler.get_chain(discard=burnin, flat=True)
ror_samples = np.exp(0.5 * samples[:, -1])
```

Dieser Code stellt die Grundlage für eine robustere Parameterschätzung mittels MCMC dar und liefert verlässliche Posterior-Verteilungen für alle Modellparameter inklusive des relativen Planetenradius.

### 5.6.6 Parameter-Umwandlung in physikalische Größen

Die Funktion `theta_to_phys(theta)` extrahiert die Parameter aus `theta` und wandelt sie in physikalisch sinnvolle Größen um. Dabei werden die logarithmischen Parameter mittels `np.exp` in lineare Werte umgewandelt: die Periode `P` in Tagen, die Transiddauer `T14` in Tagen, den Stossparameter `b` sowie die Tiefe `depth` ebenfalls in entsprechenden Einheiten.

#### Listing 6: Parameterumwandlung in physikalisch interpretierte Größen

```
def theta_to_phys(theta):
```

```

offset, rho\_sol, u1, u2, logP, t0, logT14, b\_sq, logdepth = theta
P = np.exp(logP)
T14 = np.exp(logT14)
b = np.sqrt(max(b\_sq, 0.0))
ror = np.exp(0.5 * logdepth)
depth = np.exp(logdepth)
rho = rho\_sol * RHO\_SUN
return dict(offset=offset, rho\_sol=rho, u1=u1, u2=u2, P=P, t0=t0,
            T14=T14, b=b, ror=ror, depth=depth)

```

Der Stossparameter  $b$  wird direkt aus  $b\_sq$  und die relative Bahngrösse  $ror$  aus  $logdepth$  berechnet. Die Dichte  $rho$  wird durch Multiplikation mit der Sonnen-Dichte  $RHO\_SUN$  aus der normierten Sternendichte  $rho\_sol$  bestimmt. Der Rückgabewert ist eine Tabelle mit all diesen Grössen zur weiteren Verwendung im Transitmodell.

Der Stossparameter  $b$  wird direkt aus  $b\_sq$  und die relative Bahngrösse  $ror$  aus  $logdepth$  berechnet. Die Dichte  $rho$  wird durch Multiplikation mit der Sonnen-Dichte  $RHO\_SUN$  aus der normierten Sternendichte  $rho\_sol$  bestimmt. Der Rückgabewert ist eine Tabelle mit all diesen Grössen zur weiteren Verwendung im Transitmodell.

### 5.6.7 Visualisierung der Modell-Zeitreihe

Die Funktion `plot_model_time_series` erstellt eine grafische Darstellung der Transitlichtkurve basierend auf den MCMC-Sample-Daten. Sie berechnet den Mediansatz der Parameter als repräsentativen Modellwert und generiert ein geglättetes Zeitraster für die Modellvorhersage.

Aus zufälligen Ziehungen aus den Posterior-Samples werden Vorhersagen erstellt, aus denen 16., 50. (Median) und 84. Perzentil bestimmt werden, um Unsicherheitsbereiche abzubilden.

#### Listing 7: Funktion zur Darstellung der Transitlichtkurve mit Unsicherheiten

```

def plot_model_time_series(samples, x, y, yerr, transit_model, ndraws
=200, nsmooth=2000, figsize=(8,5), save=None):
    # Median der Samples als repräsentativer Satz
    th_med = np.median(samples, axis=0)

    xgrid = np.linspace(np.min(x), np.max(x), nsmooth)

    draw_idx = np.random.default_rng(0).integers(0, len(samples), size=
ndraws)
    preds = []
    for i in draw_idx:
        try:
            preds.append(transit_model(samples[i], xgrid))
        except Exception:
            continue

    preds = np.array(preds)
    p16, p50, p84 = np.percentile(preds, [16, 50, 84], axis=0)

    # Modellwerte am Originalzeitpunkt
    ymod = transit_model(th_med, x)

    # Chi-Quadrat und reduziertes Chi-Quadrat zur Bewertung des Modells
    chi2 = np.sum(((y - ymod)/yerr)**2)
    dof = max(len(y) - samples.shape[1], 1)

```

```

chi2_red = chi2 / dof

# Plot erstellen
fig, ax = plt.subplots(2, 1, figsize=figsize, sharex=True,
    gridspec_kw={'height_ratios':[3,1]})
ax[0].errorbar(x, y, yerr, fmt='.', alpha=0.6, label='data')
ax[0].plot(xgrid, p50, lw=2, label='model_□(median)')
ax[0].fill_between(xgrid, p16, p84, alpha=0.25, label='model_□
    plus_minus_1_sigma')
ax[0].set_ylabel('Flux')
ax[0].legend(loc='best', frameon=False)
ax[0].set_title(rf"Best_□model;_□$\chi^2_{\mathrm{{red}}}_\square\approx_\square{
    chi2\_red:.2f}$")

ax[1].axhline(0, ls='--')
ax[1].errorbar(x, y - ymod, yerr, fmt='.', alpha=0.6)
ax[1].set_xlabel('Time_□[days]')
ax[1].set_ylabel('Residual')

plt.tight_layout()

if save:
    plt.savefig(save, dpi=200, bbox_inches='tight')
plt.show()

```

Zur Modellbewertung werden die Chi-Quadrat-Statistik und das reduzierte Chi-Quadrat berechnet. Die obere Grafik zeigt die beobachteten Daten mit Fehlerbalken, den Median-Modellverlauf und den Unsicherheitsbereich. Die untere Grafik präsentiert die Residuen (Differenz von Daten und Modell) inklusive Fehlerbalken. Optional kann das Diagramm gespeichert werden.

### 5.6.8 Transformation der MCMC-Samples und Erstellung des Corner Plots

Die Funktion `corner_transforms` transformiert die Rohdaten der MCMC-Ketten in für die Visualisierung geeignete Parameter. Dabei werden logarithmisch gespeicherte Werte wie Periode  $\log P$  und Transitdauer  $\log T_{14}$  in lineare Skalen zurückgeführt. Der Impact-Parameter  $b$  wird als Quadratwurzel aus dem quadrierten Wert abgeleitet. Aus dem logarithmischen Wert der Transittiefe wird das Planeten-Stellen-Verhältnis  $\text{ror}$  sowie die Transit-Tiefe `depth` berechnet.

Die transformierten Größen werden zu einer Matrix zusammengefügt, die gemeinsam mit einer Liste von Achsenbeschriftungen für die Erstellung des Corner-Plots dient.

#### Listing 8: Parametertransformation und Corner Plot

```

def corner_transforms(samples):
    P = np.exp(samples[:, 4])
    T14 = np.exp(samples[:, 6])
    b = np.sqrt(np.clip(samples[:, 7], 0.0, None))
    ror = np.exp(0.5 * samples[:, 8])
    depth = np.exp(samples[:, 8])
    out = np.column_stack([
        samples[:, 0], # offset / Durchschnitt
        samples[:, 1], # rho_star [rho_sun]
        samples[:, 2], # u1
        samples[:, 3], # u2
        P, # period [d]

```

```

    samples[:, 5], # t0 [d]
    T14, # Transitdauer [d]
    b, # Stossparameter
    ror, # Rp/R*
    depth # tiefe (einheitslos)
])
labels = [
    r'offset', r'\rho_\star/\rho_\odot$', r'$u_1$', r'$u_2$', r'$P$□[d]
    ',
    r'$t_0$□[d]', r'$T_{14}$□[d]', r'$b$', r'$R_p/R_\star$', 'depth'
]
return out, labels

def plot_corners(samples, max_points=50000, save=None):
    data, labels = corner_transforms(samples)
    if len(data) > max_points:
        rng = np.random.default_rng(42)
        idx = rng.choice(len(data), size=max_points, replace=False)
        data = data[idx]
    fig = corner.corner(
        data, labels=labels, show_titles=True,
        quantiles=[0.16, 0.50, 0.84], title_fmt=".4f"
    )
    if save:
        fig.savefig(save, dpi=200, bbox_inches='tight')

```

Die Funktion `plot_corners` beschränkt gegebenenfalls die Anzahl der Datenpunkte durch Zufallsstichprobe auf `max_points` und erzeugt mit dem `corner` Paket die graphische Darstellung mit Titeln und Konfidenzbereichen. Optional kann der Plot als Bilddatei gespeichert werden.

### 5.6.9 Ausgabe und Visualisierung der Ergebnisse

Im letzten Schritt der Analyse werden die Ergebnisse der MCMC-Simulationen durch verschiedene grafische Darstellungen ausgegeben, die einen umfassenden Einblick in die Güte und Streuung der Modelle ermöglichen.

Die Funktion `plot_model_time_series` visualisiert den zeitlichen Verlauf der Transitlichtkurve zusammen mit Unsicherheitsbereichen, berechnet aus zufälligen Ziehungen der Posterior-Verteilung. Dieses Diagramm zeigt sowohl die gemessenen Daten als auch das modellierte Signal über die Beobachtungszeit.

Schliesslich erzeugt `plot_corners` einen Corner Plot, welcher die Verteilungen der modellbasierten Parameter inklusive ihrer Korrelationen mit quantilenbasierten Unsicherheitsgrenzen grafisch aufbereitet. Dies unterstützt die Evaluierung der Parameterschätzungen und möglicher Parameterabhängigkeiten.

Der Aufruf von `plt.show()` stellt sicher, dass alle Plots gezeichnet und angezeigt werden.

#### Listing 9: Visualisierung und Ausgabe der MCMC-Ergebnisse

```

plot_model_time_series(samples, x, y, yerr, transit_model, ndraws=200,
    save=None)
plot_corners(samples, max_points=40000, save="corner.png")
plt.show()

```

Diese Visualisierungsschritte sind essenziell für die Interpretation des Modellfits und die Kommunikation der Analyseergebnisse in der wissenschaftlichen Arbeit.

## 5.7 Gemeinsamer Multi-Instrument Transit-Fit

Um die Transitparameter von TOI-7265.01 so präzise und robust wie möglich zu bestimmen, wurden die Lichtkurven aller drei Beobachtungsinstrumente simultan in einem gemeinsamen Fit (engl. *joint fit*) analysiert. Die drei Datensätze sind:

- Saint-EX: 534 Datenpunkte, Beobachtung vom 2. Juli 2025.
- Lyceum 130 Observatory, Nowosibirsk (LHO): 62 Datenpunkte, Beobachtung vom 15. Oktober 2025.
- TESS: 11 766 Datenpunkte nach GP-Bereinigung mit Nuance.

Durch den simultanen Fit werden alle Datensätze gemeinsam auf denselben Parametersatz angepasst, was systematische Fehler reduziert und die Statistik erheblich verbessert.

### 5.7.1 Parametervektor und Prior-Verteilungen

Der gemeinsame Parametervektor umfasst 19 Dimensionen: 3 globale Parameter, die für alle Datensätze gelten, sowie je 5 instrumentspezifische Parameter pro Datensatz (3 Datensätze  $\times$  5 = 15 Parameter). Die globalen Parameter beschreiben die fundamentalen Transiteigenschaften, während die instrumentspezifischen Parameter instrumentelle Unterschiede (Rp/Rs, Limb-Darkening, Jitter, Baseline) modellieren.

**Tabelle 2:** Parametervektor des gemeinsamen Transit-Fits (19 Dimensionen)

Nr.	Parameter	Beschreibung	Prior
<i>Globale Parameter (4)</i>			
1	$t_0$	Transitmitte (BJD)	Normalverteilung ( $\mu, \sigma$ )
2	$\ln(\rho_*)$	Log. Sterndichte [ $\text{g/cm}^3$ ]	Gleichverteilt $[-2, 4]$
3	$r_1$	Espinoza- $r_1$ (Impact/Radius)	Gleichverteilt $[0, 1]$
4	$r_2$	Espinoza- $r_2$ (Impact/Radius)	Gleichverteilt $[0, 1]$
<i>SaintEX-spezifisch (5)</i>			
5	$(r_1)_{\text{SX}}$	Espinoza- $r_1$ für SaintEX	Gleichverteilt $[0, 1]$
6	$(r_2)_{\text{SX}}$	Espinoza- $r_2$ für SaintEX	Gleichverteilt $[0, 1]$
7	$(q_1)_{\text{SX}}$	Kipping- $q_1$ Limb-Darkening	Gleichverteilt $[0, 1]$
8	$(q_2)_{\text{SX}}$	Kipping- $q_2$ Limb-Darkening	Gleichverteilt $[0, 1]$
9	$\ln(\sigma_{w,\text{SX}})$	Log. Jitter SaintEX	Gleichverteilt $[-15, 0]$
<i>LHO Nowosibirsk-spezifisch (5)</i>			
10	$(r_1)_{\text{LHO}}$	Espinoza- $r_1$ für LHO	Gleichverteilt $[0, 1]$
11	$(r_2)_{\text{LHO}}$	Espinoza- $r_2$ für LHO	Gleichverteilt $[0, 1]$
12	$(q_1)_{\text{LHO}}$	Kipping- $q_1$ Limb-Darkening	Gleichverteilt $[0, 1]$
13	$(q_2)_{\text{LHO}}$	Kipping- $q_2$ Limb-Darkening	Gleichverteilt $[0, 1]$
14	$\ln(\sigma_{w,\text{LHO}})$	Log. Jitter LHO	Gleichverteilt $[-15, 0]$
<i>TESS-spezifisch (5)</i>			
15	$(r_1)_{\text{TESS}}$	Espinoza- $r_1$ für TESS	Gleichverteilt $[0, 1]$
16	$(r_2)_{\text{TESS}}$	Espinoza- $r_2$ für TESS	Gleichverteilt $[0, 1]$
17	$(q_1)_{\text{TESS}}$	Kipping- $q_1$ Limb-Darkening	Gleichverteilt $[0, 1]$
18	$(q_2)_{\text{TESS}}$	Kipping- $q_2$ Limb-Darkening	Gleichverteilt $[0, 1]$
19	$\ln(\sigma_{w,\text{TESS}})$	Log. Jitter TESS	Gleichverteilt $[-15, 0]$

### 5.7.2 Log-Likelihood-Funktion

Die Qualität einer Parameterauswahl wird über die logarithmierte Likelihood-Funktion bewertet. Für jeden Datenpunkt  $i$  mit gemessenem Fluss  $y_i$ , Modellfluss  $m_i(\theta)$ , Mesunsicherheit  $\sigma_i$  und instrumentspezifischem Jitter  $\sigma_w$  gilt:

$$\ln \mathcal{L}(\theta) = -\frac{1}{2} \sum_i \left[ \frac{(y_i - m_i)^2}{\sigma_i^2 + \sigma_w^2} + \ln(2\pi(\sigma_i^2 + \sigma_w^2)) \right]$$

Der Jitter-Term  $\sigma_w^2$  wird zur gemessenen Unsicherheit  $\sigma_i^2$  addiert und modelliert zusätzliches weisses Rauschen, das nicht durch die formalen Photometriefehler erfasst wird (z. B. atmosphärisches Szintillationsrauschen, Instrumentenrauschen). Der  $\ln$ -Term gewährleistet die korrekte Normierung und bestraft unnötig grosse Jitter-Werte.

Das Transitmodell  $m_i(\theta)$  wird für jeden Datenpunkt mit der Python-Bibliothek BATMAN (Basic Transit Model Calculation) berechnet, die analytische Lichtkurvenmodelle nach Mandel & Agol (2002) implementiert (Kreidberg, 2015).

### 5.7.3 MCMC-Konfiguration und Codestruktur

Der MCMC-Sampler basiert auf dem Ensemble-Sampler emcee (Foreman-Mackey et al., 2013), der den affine-invarianten Stretch-Move-Algorithmus nach Goodman & Weare (2010) imple-

mentiert. Die Konfiguration des Samplers ist:

- Anzahl Walkers: 100 parallele Ketten (mindestens  $2 \times$  Parameterdimension)
- Anzahl Schritte: 20 000 Schritte pro Walker
- Burn-in: 65 % der Schritte werden als Einschwingphase verworfen (32 500 Schritte)
- Genutzte Samples:  $100 \times 7\,000 = 700\,000$  Posterioren-Samples

Die Implementierung ist in der Datei `transitworker.py` strukturiert, welche die Funktion `log_posterior(theta)` enthält. Diese Funktion ruft zunächst den Prior auf – bei unzulässigen Parametern wird  $-\infty$  zurückgegeben – und berechnet dann für alle drei Datensätze die Log-Likelihood, die aufsummiert wird. Zur Beschleunigung der Berechnung wird Python's multiprocessing verwendet, sodass die 150 Walker-Evaluationen parallel auf mehreren CPU-Kernen ausgeführt werden (Espinoza, 2018; Kipping, 2013). Für die zeitliche Ersparnis wurde die MCMC Berechnung auf einem externen Server mit einem AMD Ryzen Threadripper 32-Core CPU und 32 Gigabyte Arbeitsspeicher ausgeführt.

Die Python Implementierung ist im Anhang dieser Arbeit zu finden, da eine Erläuterung des Codes in der Methodik von keinem Nutzen wäre, da alle Prozesse bereits thematisiert wurden.

## 6 Resultate und Diskussion

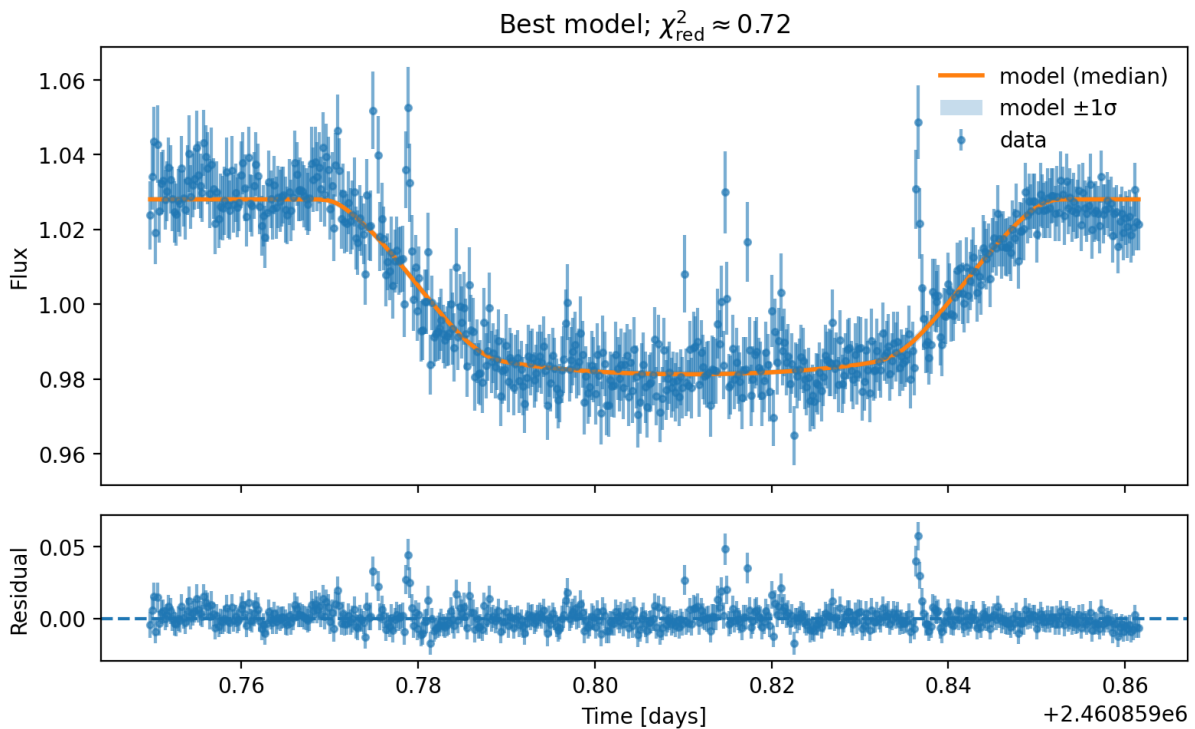
In diesem Kapitel werden die Resultate der Modellanpassung und der Parameterabschätzung anhand visueller und quantitativer Methoden dargestellt und interpretiert.

### 6.1 Transitmodell und Residuenplot anhand der Eigenbeobachtungen

Die mit dem Saint-Ex-Observatorium aufgezeichneten Beobachtungsdaten wurden mit dem selbst programmierten Python-Skript ausgewertet. Aus diesen Daten wurde ein Transitmodell erstellt, aus welchem dann die Exoplanetparameter bestimmt wurden. In der Abbildung 19 ist das gefittete Modell ersichtlich.

Abbildung 19 visualisiert den Vergleich der Messdaten (blaue Punkte mit Fehlerbalken) und dem mittels MCMC bestimmten Modell (orange Linie), begleitet vom  $1\sigma$ -Unsicherheitsbereich. Die Residuen (untere Grafik der Abbildung 19) verteilen sich zufällig um null herum und zeigen weder einen klaren Trend noch eine systematische Struktur. Das Reduzierte  $\chi^2$  beträgt ungefähr 0.72, was eine gute Modellanpassung signalisiert (Singal & Rana, 2015).

**Abbildung 19:** Transitmodell und Residuenplot von TOI 7265.01



## 6.2 Corner-Plot der Saint-Ex-Daten

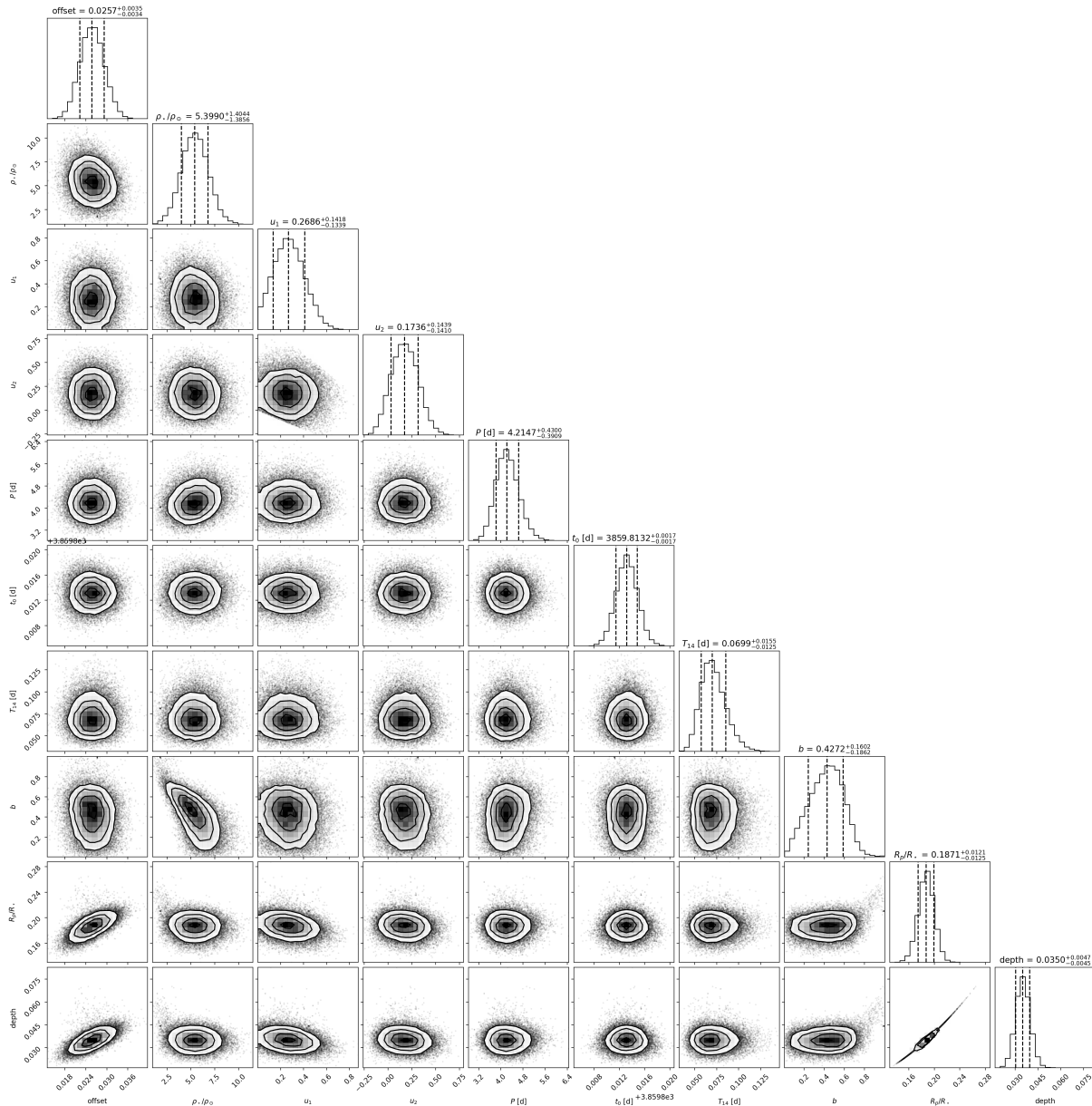
Abbildung 20 zeigt die Exoplanetparameter, wie sie für die Saint-Ex-Transitbeobachtungen aus der MCMC-Analyse mit nur dem Saint-Ex-Datensatz resultieren.

Folgend werden alle berechneten Exoplanetparameter aufgelistet:

- Offset:  $0.0257^{+0.0035}_{-0.0034}$
- Normalisierte Sternendichte  $\rho_*/\rho_\odot$ :  $5.3990^{+1.4044}_{-1.3856}$
- Randverdunklungskoeffizient 1  $u_1$ :  $0.2686^{+0.1418}_{-0.1339}$
- Randverdunklungskoeffizient 2  $u_2$ :  $0.1736^{+0.1439}_{-0.1410}$
- Periode  $P$ :  $4.2147^{+0.4300}_{-0.3909}$  Tage
- Transitmitte  $t_0$ :  $3859.8132^{+0.0017}_{-0.0017}$  Tage
- Transiddauer  $T_{14}$ :  $0.0699^{+0.0155}_{-0.0125}$  Tage
- Impact-Parameter  $b$ :  $0.4272^{+0.1602}_{-0.1862}$
- Verhältnis  $R_p/R_*$ :  $0.1871^{+0.0121}_{-0.0125}$
- Transittiefe:  $0.0350^{+0.0047}_{-0.0045}$

Im Corner-Plot, welcher in Abbildung 20 zu erkennen ist, befindet sich jeweils an der rechten Seite der Grafik eine Verteilungsfigur des jeweiligen Parameters. Wenn diese Figur visuell annähernd normalverteilt ist, dann handelt es sich um ein gut angepasstes Transitmodell (Tuchow et al., 2019). Dies kann auch grob visuell bestätigt werden und zwar indem man das in Abbildung 19 aufgeführte Transitmodell mit den Observationsdaten vergleicht und so erkennt, dass keine signifikante Abweichung vorhanden ist.

**Abbildung 20:** Corner-Plot der Parameter für die Saint-Ex-Transitdaten



### 6.3 Corner-Plot der TESS-Daten

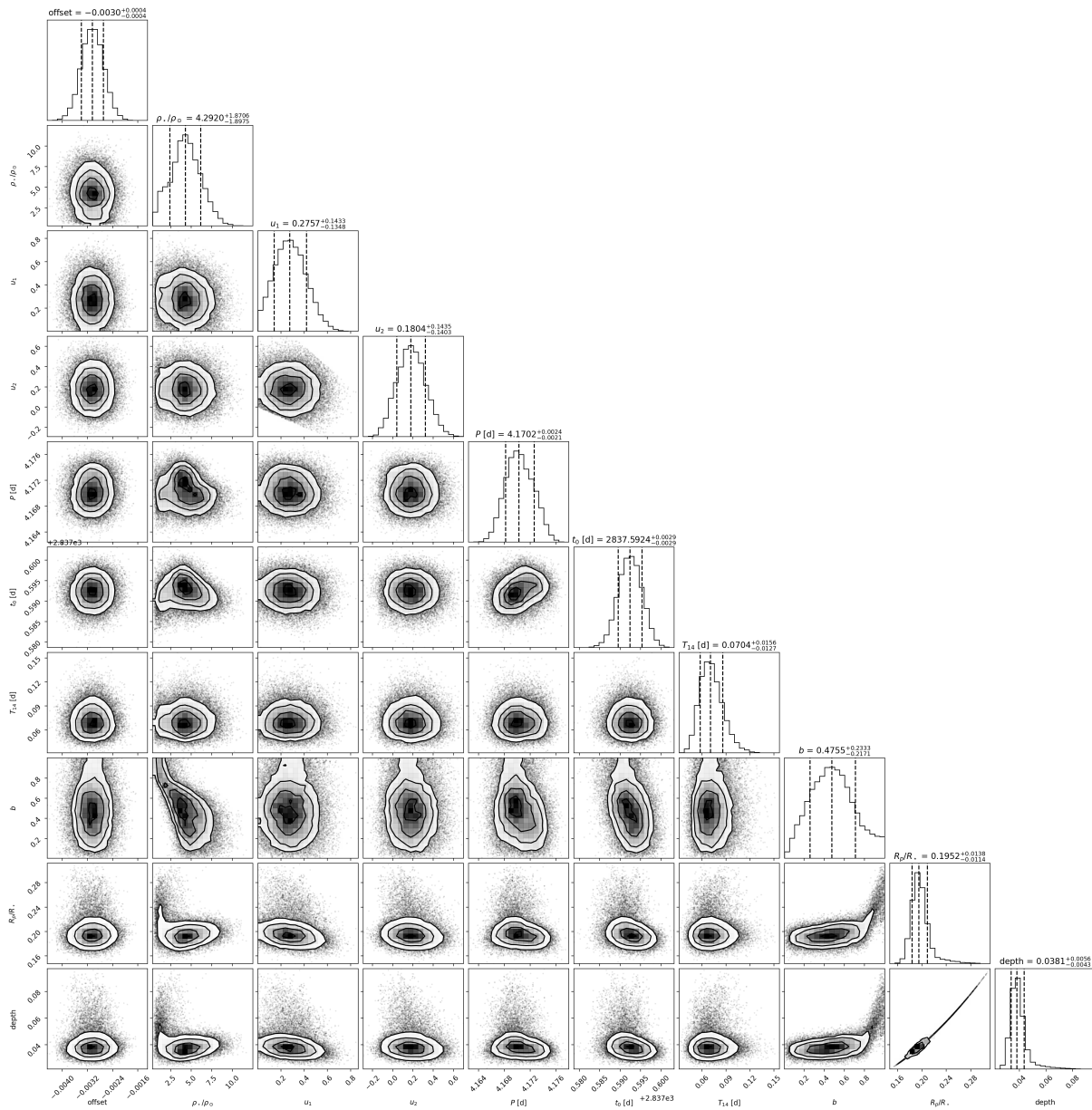
Abbildung 21 zeigt die Exoplanetparameter, wie sie für die TESS-Transitbeobachtungen aus der MCMC-Analyse mit nur den TESS-Daten resultieren. Diese Daten wurden verwendet, um die Plausibilität der Daten, welche durch das Saint-Ex-Observatorium gemessen wurden, zu prüfen.

Folgend werden alle berechneten Exoplanetparameter aufgelistet:

- Offset:  $0.0030^{+0.0004}_{-0.0004}$
- Normalisierte Sternendichte  $\rho_*/\rho_\odot$ :  $4.2920^{+1.8706}_{-1.8975}$
- Randverdunklungskoeffizient 1  $u_1$ :  $0.2757^{+0.1433}_{-0.1348}$
- Randverdunklungskoeffizient 2  $u_2$ :  $0.1804^{+0.1435}_{-0.1403}$
- Periode  $P$ :  $4.1702^{+0.0024}_{-0.0021}$  Tage
- Transitmitte  $t_0$ :  $2837.5924^{+0.0029}_{-0.0029}$  Tage
- Transiddauer  $T_{14}$ :  $0.0704^{+0.0156}_{-0.0127}$  Tage
- Impact-Parameter  $b$ :  $0.4755^{+0.2333}_{-0.2171}$
- Verhältnis  $R_p/R_*$ :  $0.1952^{+0.0138}_{-0.0114}$
- Transittiefe:  $0.0381^{+0.0056}_{-0.0043}$

In Abbildung 21 sind erneut annähernd normalverteilte Verteilungsfiguren zu erkennen, auch um vernünftige Exoplanetparameter (Tuchow et al., 2019).

**Abbildung 21:** *Corner-Plot der Parameter für die TESS-Transitdaten.*



## 6.4 Vergleich von Saint-Ex- und TESS-Daten

Der direkte Vergleich der Exoplanetenparameter aus den unabhängigen Saint-Ex- und TESS-Transitbeobachtungen zeigt eine bemerkenswerte Übereinstimmung in allen wesentlichen Größen, wie Periode, Transitdauer, Stossparameter, Planeten-zu-Stern-Radiusverhältnis und Transittiefe. Die Fehlerbereiche des Corner-Plots überlappen sich weitgehend, was auf eine hohe Konsistenz der Daten und der Modellanpassungen hinweist.

Diese hohe Ähnlichkeit stärkt massgeblich die Evidenz des Signals und bestätigt mit grosser Wahrscheinlichkeit die Existenz des Exoplaneten TOI 7265.01.

## 6.5 Resultate des gemeinsamen Multi-Instrument-Fits

Nach Konvergenz des MCMC-Samplers mit 100 Walkern über 20 000 Schritte (65% Burn-in) stehen 700 000 unabhängige Posterioren-Samples für die Parameterauswertung zur Verfügung. Die Mediane der Marginalverteilungen werden als Punktschätzer verwendet, die 16. und 84. Perzentile als  $1\sigma$ -Konfidenzintervalle.

### 6.5.1 Globale Transitparameter

Die in Tabelle 3 dargestellten globalen Transitparameter ergeben sich aus einem gemeinsamen Multi-Instrument-Fit für TOI-7265.01.

**Tabelle 3:** Globale Transitparameter des gemeinsamen Multi-Instrument-Fits für TOI-7265.01

Parameter	Symbol	Wert	Einheit
Transitmitte	$T_0$	$2\,460\,576,09476 \pm 0,00025$	BJD
Sterndichte	$\rho_*$	$6,368^{+0,424}_{-0,380}$	$\text{g/cm}^3$
Halbachsenverhältnis	$a/R_*$	$18,029^{+0,391}_{-0,366}$	–
Inklination	$i$	$88,63^{+0,25}_{-0,22}$	Grad
Stossparameter	$b$	$0,432^{+0,063}_{-0,072}$	–
Radiusverhältnis (kombi.)	$R_p/R_s$	$0,19744 \pm 0,00478$	–
Planetenradius	$R_p$	$10,52 \pm 0,25$	$R_\oplus$
Planetenradius	$R_p$	$0,938 \pm 0,023$	$R_{\text{Jup}}$
Transitdauer	$T_{14}$	$1,976 \pm 0,03$	Stunden
Gleichgewichtstemperatur	$T_{\text{eq}}$	$564,9 \pm 6,0$	K
Einstrahlungsfluss	$S_{\text{inc}}$	$16,9 \pm 0,71$	$S_\oplus$

### 6.5.2 Instrumentspezifische Radiusverhältnisse und Gütemasse

Jedes Instrument verfügt über ein eigenes Radiusverhältnis  $R_p/R_s$ , da unterschiedliche Bandpässe und Apertur-Konfigurationen leicht verschiedene Transitiefen messen können. Der kombinierte  $R_p/R_s$ -Wert wird aus dem Gesamtposterior bestimmt.

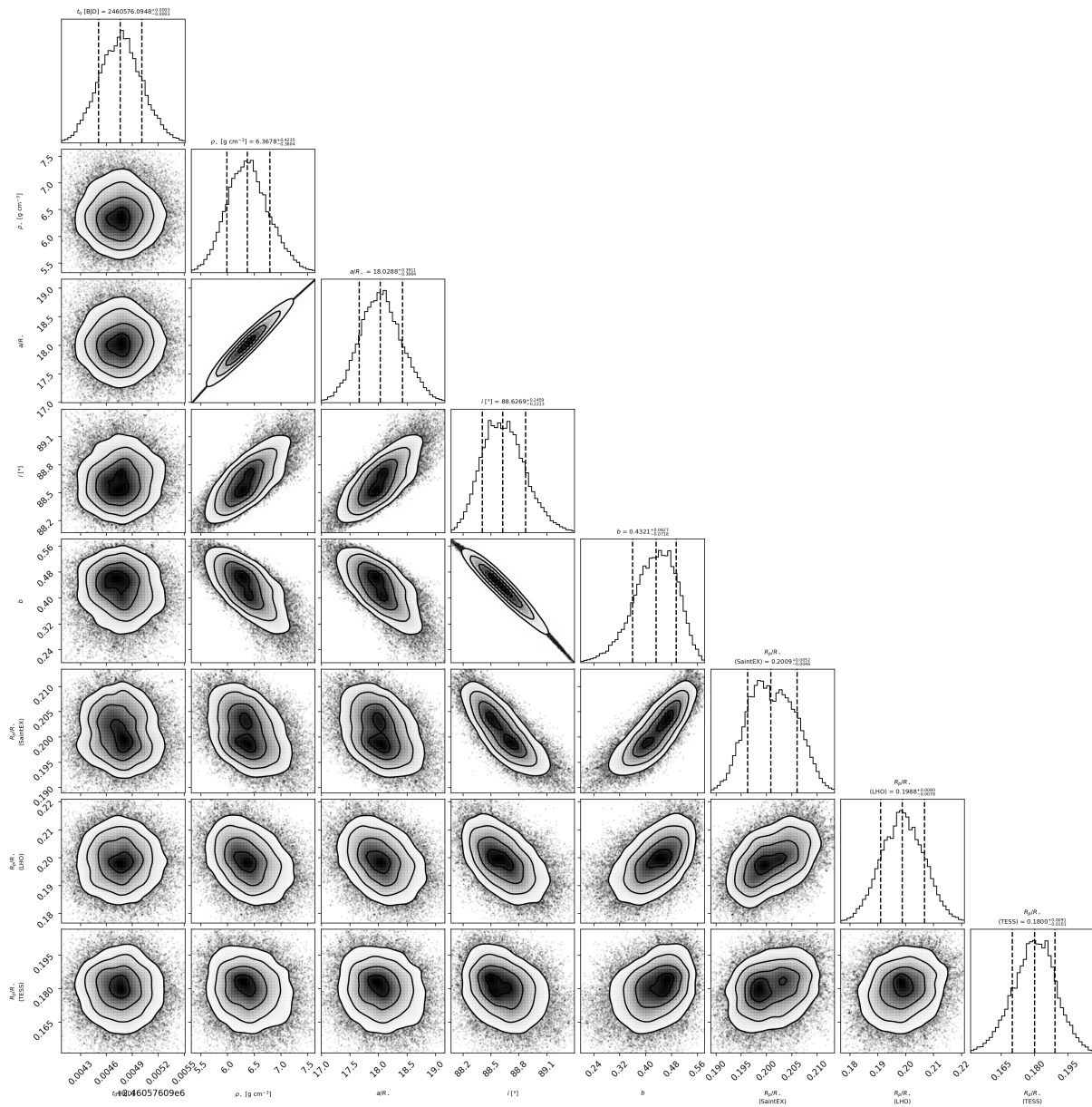
**Tabelle 4:** Instrumentspezifische Radiusverhältnisse und reduziertes Chi-Quadrat ( $\chi^2_\nu$ )

Instrument	Datenpunkte	$R_p/R_s$	$\chi^2_\nu$
Saint-EX	534	$0,20087 \pm 0,005$	1,07
LHO Nowosibirsk	62	$0,19877 \pm 0,008$	1,23
TESS	11 766	$0,18004 \pm 0,010$	1,01

### 6.5.3 Corner-Plot der Posterioren

Der Corner-Plot in Abbildung 22 zeigt die zweidimensionalen Marginalverteilungen der 9 wichtigsten Parameter: Transitmitte  $t_0$ , Sterndichte  $\rho_*$ , Halbachsenverhältnis  $a/R_*$ , Inklination  $i$ , Stossparameter  $b$  sowie die drei instrumentspezifischen Radiusverhältnisse  $R_p/R_s$  für Saint-EX, LHO und TESS. Alle Marginalverteilungen sind annähernd gaussisch und gut lokalisiert, was auf eine erfolgreiche Konvergenz des Samplers und auf die Wirksamkeit der verwendeten Reparametrisierungen nach Espinoza (2018) und Kipping (2013) hinweist (Espinoza, 2018;

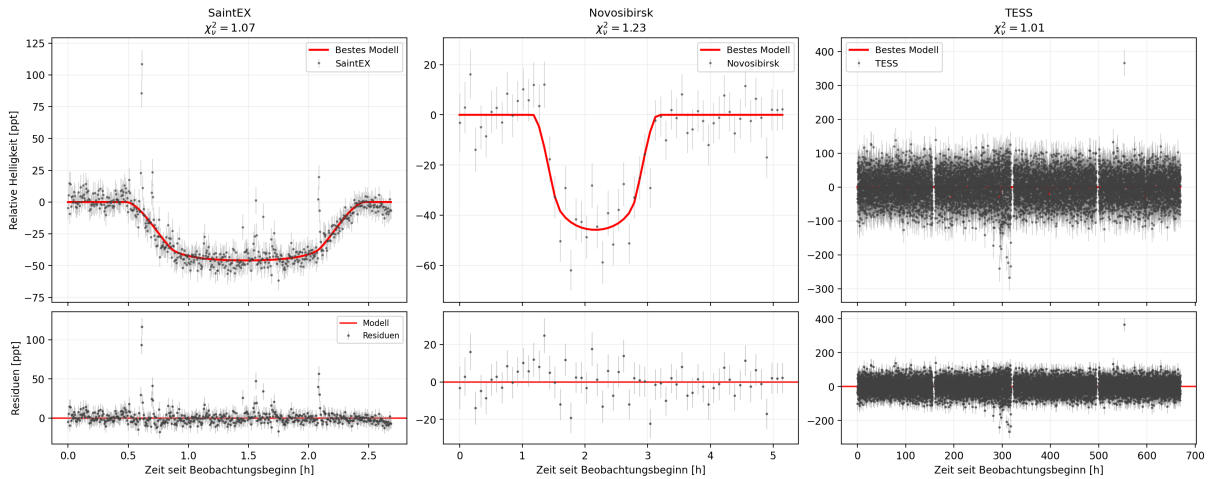
Kipping, 2013). Keine starken unerklärbaren Korrelationen oder degenerierte Strukturen sind sichtbar und somit sind die Parameter des Joint-Fit-Modells gültig.



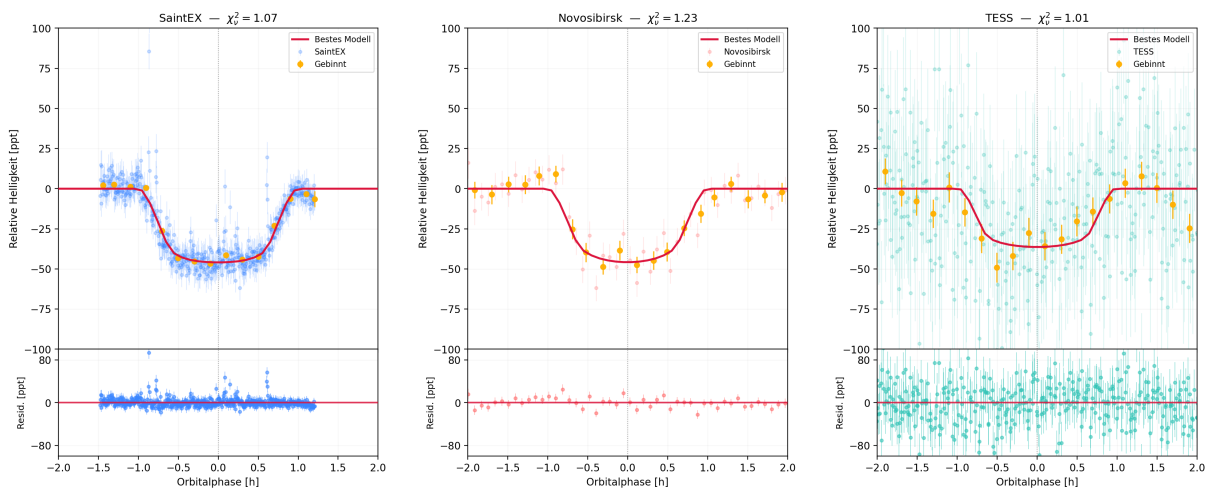
**Abbildung 22:** Corner-Plot der Posteriors des gemeinsamen Multi-Instrument-Fits. Die neun dargestellten Parameter zeigen annähernd gauss'sche, dekorrelierte Verteilungen.

### 6.5.4 Phasengefaltete Lichtkurven und Transitmodell

Abbildung 23 zeigt das beste Transitmodell (Median-Posterior) überlagert auf die Rohdaten aller drei Instrumente. Abbildung 24 zeigt die phasengefaltete Gesamtlichtkurve aller Datensätze kombiniert, phasiert auf die mittlere Periode und Transitmitte. Die Phase null entspricht dem Zentrum des Transits.



**Abbildung 23:** Das Transitmodell (rote Linie) überlagert auf die Beobachtungsdaten aller drei Instrumente. Die Residuen sind im unteren Panel dargestellt.

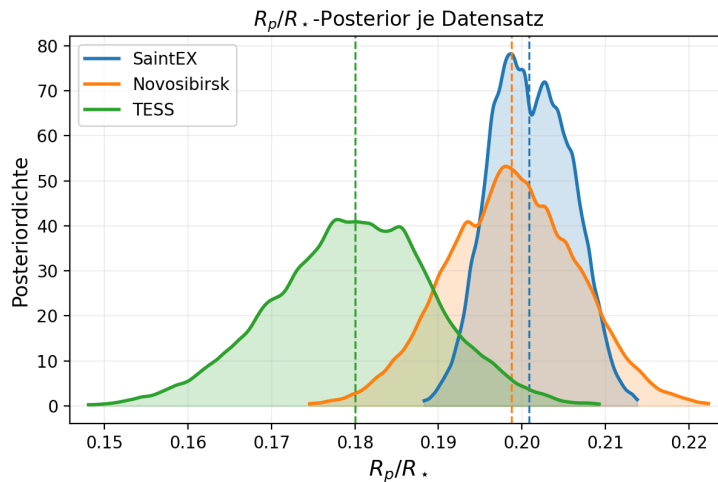


**Abbildung 24:** Phasengefaltete Lichtkurve aller drei Datensätze, zentriert auf die Transitmitte ( $\phi = 0$ ). Die Datenpunkte sind farblich nach Instrumente kodiert, die rote Linie zeigt das Medianmodell.

Man erkennt gut in beiden Abbildungen, wie das Joint-Fit-Transit-Modell gut auf alle Datensätze passt. Der  $\chi^2_\nu$ -Wert liegt bei allen Datensätzen ungefähr bei 1. Dies steht für ein sehr gut passendes Modell (Singal & Rana, 2015).

### 6.5.5 Posterioren-Verteilung der Radiusverhältnisse

Die in Abbildung 25 dargestellte Kerndichteschätzung (KDE) der Posteriorverteilung des Radiusverhältnisses  $R_p/R_s$  ist zentral für die physikalische Charakterisierung von TOI-7265.01. Der Median (senkrechte Linie) quantifiziert den wahrscheinlichsten Wert und die zugehörige Unsicherheit des Radiusverhältnisses, das direkt in den abgeleiteten Planetenradius und damit in Dichte, Zusammensetzung und Atmosphäreigenschaften eingeht. Eine unimodale, gut definierte Verteilung ohne ausgeprägte Nebenmaxima spricht dafür, dass das Transitmodell die Daten konsistent beschreibt und keine starken Degenerenzen mit anderen Parametern vorliegen.



**Abbildung 25:** Kerndichteschätzung (KDE) der Posterioren-Verteilung des Radiusverhältnisses  $R_p/R_s$ .

Gleichzeitig erlaubt die Form der Posterioren-Verteilung einen direkten Vergleich mit unabhängigen Datensätzen und Instrumenten: Nur wenn sich die jeweiligen Posterioren überlappen und kompatible Mediane liefern, kann eine robuste Planetendetektion und -validierung behauptet werden. Somit bildet dieser Graph einen essenziellen Schritt, um systematische Effekte auszuschliessen und die Zuverlässigkeit der aus den Transits abgeleiteten Planetenparameter zu belegen. Zu beachten ist ausserdem, dass trotz der schlechteren Auflösung der TESS-Daten es trotzdem eine Überlappung der Posterior-Verteilungen gibt. Somit kann durch die Überlappung der Posterior-Verteilungen also die Validation der Planeteneigenschaft von TOI-7265.01 stammen, da verschiedene Observationen mit unterschiedlichen Instrumenten dasselbe erkennen!

## 6.6 Scheiternde Statistische Validierung mittels TRICERATOPS

TRICERATOPS wurde bisher in der Methodik nicht erwähnt. Dies liegt daran, dass die wissenschaftlich gängige Exoplanetvalidation für TOIs über TRICERATOPS funktioniert und Leider scheitert diese Rechnung an den physikalischen Gegebenheiten des Sternsystems von TOI-7265.01. Da alle anderen Validierungsprogramme sich auch auf TESS-Daten beruhen oder TOI-7265.01 nicht abdecken, konnte keine einfache Lösung verwendet werden. TRICERATOPS (Giacalone et al., 2021) ist ein weit verbreitetes bayesianisches Werkzeug zur statistischen Validierung von TESS-Planetenkandidaten. Für ein gegebenes Transitsignal berechnet es die False Positive Probability (FPP) sowie die Nearby False Positive Probability (NFPP), indem es die Wahrscheinlichkeiten verschiedener astrophysikalischer Szenarien gegeneinander abwägt. Diese Szenarien sind ein transitierender Planet (TP), ein eklipsierender Doppelstern (EB) oder ein Hintergrund-EB (BEB) auf dem Zielstern selbst, sowie analoge Szenarien auf aufgelösten Nachbarsternen (NTP, NEB).

Ein Kandidat gilt als *statistisch validiert*, wenn  $FPP < 0,015$  und  $NFPP < 10^{-3}$  gleichzeitig erfüllt sind (Giacalone et al., 2021).

Der Code für diese Analyse befindet sich im Anhang, da TRICERATOPS für diese Arbeit durch die auftretende Unzuverlässigkeit aufgrund der Konstellation des stellaren Systeme um TOI-7265.01 unbrauchbar ist.

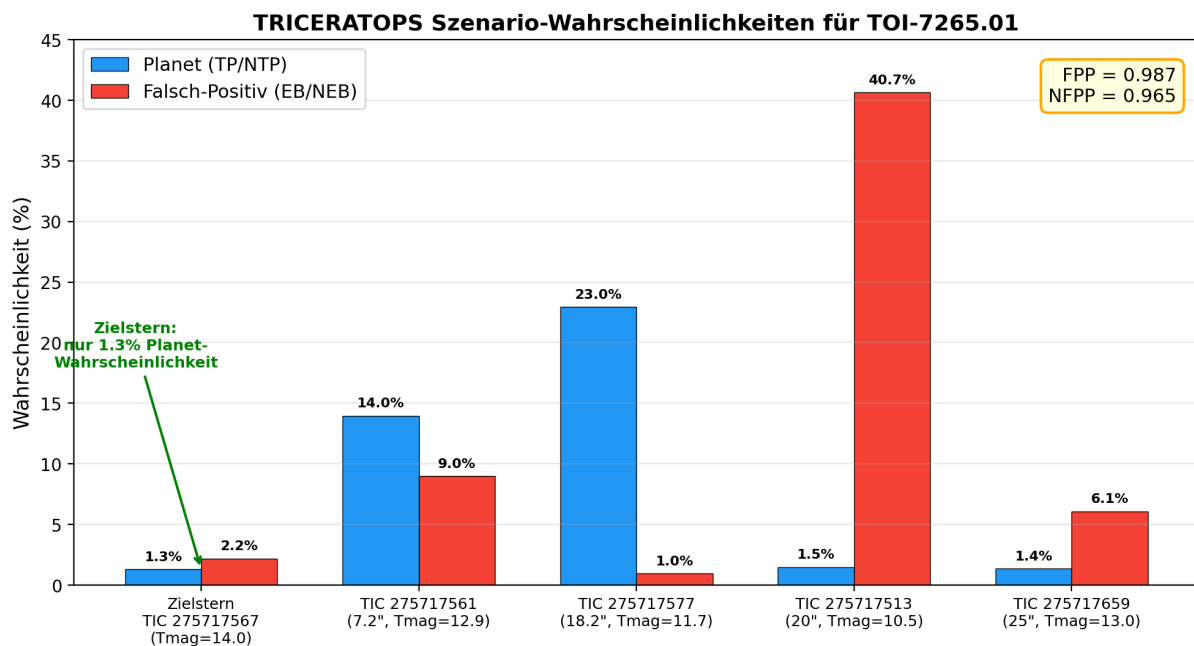
## 6.6.1 Ergebnis der TRICERATOPS-Analyse

Die TRICERATOPS-Analyse für TOI-7265.01 ergibt:

$$\text{FPP} = 0,987 \pm 0,001, \quad ()$$

$$\text{NFPP} = 0,965 \pm 0,002. \quad ()$$

Beide Werte liegen weit über den Validierungsschwellen. Die Aufschlüsselung der Szenario-Wahrscheinlichkeiten (Abbildung 26) zeigt, dass 96,5 % der Gesamtwahrscheinlichkeit auf Szenarien entfallen, bei denen das Transitsignal von einem aufgelösten Nachbarstern stammt. Lediglich 1,3 % entfallen auf das Szenario eines Planeten auf dem Zielstern TIC 275717567.



**Abbildung 26:** TRICERATOPS-Szenario-Wahrscheinlichkeiten für TOI-7265.01. Der Zielstern TIC 275717567 erhält nur 1,3 % Planetenwahrscheinlichkeit, da TRICERATOPS den Grossteil der Wahrscheinlichkeit den helleren Nachbarsternen zuordnet, die innerhalb der TESS-Apertur liegen.

## 6.6.2 Ursache: Verfälschung des Lichtflusses durch die TESS-Apertur

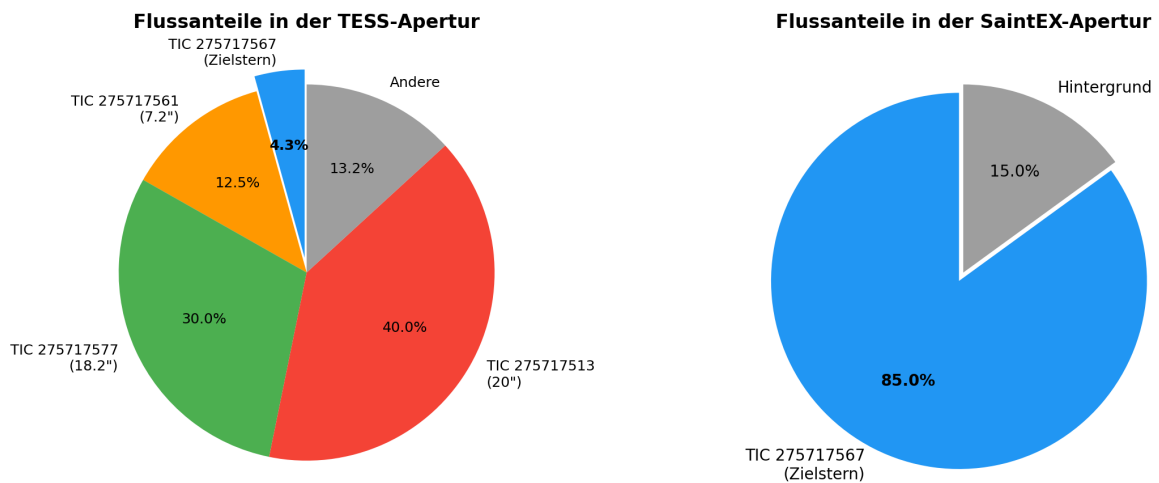
Der Grund für dieses Ergebnis liegt in der groben Pixelskala von TESS (21 Bogensekunden pro Pixel). Der Zielstern TIC 275717567 ( $T_{\text{mag}} = 14,0$ ) befindet sich in einem dicht besiedelten Sternfeld mit mehreren deutlich helleren Nachbarsternen innerhalb der photometrischen Apertur (Tabelle 5).

Abbildung 27 veranschaulicht das Problem: Während der Zielstern in der TESS-Apertur lediglich 4,3 % des Gesamtflusses beiträgt, dominiert er in der SaintEX-Apertur (0,34 Bogensekunden pro Pixel) mit 85 % des Flusses. TRICERATOPS bewertet ausschliesslich die TESS-Daten und schliesst aus dem geringen Flussanteil des Zielsterns, dass es wesentlich wahrscheinlicher ist, dass das Transitsignal von einem der helleren Nachbarsterne stammt.

**Tabelle 5:** Dominante Nachbarsterne innerhalb der TESS-Apertur von TOI-7265 und ihre TRICERATOPS-Wahrscheinlichkeiten. Der Zielstern trägt nur 4,3% des Gesamtflusses bei.

Stern	Abstand	$T_{\text{mag}}$	Flussanteil	TRICERATOPS-Prob.
TIC 275717567 (Ziel)	0,0 "	14,0	4,3 %	3,5 %
TIC 275717561	7,2 "	12,9	12,5 %	23,0 %
TIC 275717577	18,2 "	11,7	30,0 %	23,9 %
TIC 275717513	20,0 "	10,5	40,0 %	42,1 %
TIC 275717659	25,0 "	13,0	—	7,5 %

**Flussverdünnung: TESS (21"/Pixel) vs. SaintEX (0.34"/Pixel)**



**Abbildung 27:** Flussanteile in der TESS-Apertur (links) im Vergleich zur SaintEX-Apertur (rechts). In der TESS-Apertur trägt der Zielstern nur 4,3% des Gesamtflusses bei, was TRICERATOPS dazu veranlasst, die Wahrscheinlichkeit den helleren Nachbarsternen zuzuordnen.

### 6.6.3 Warum TRICERATOPS für diesen Kandidaten nicht anwendbar ist

TRICERATOPS ist als rein TESS-basiertes Werkzeug konzipiert und kann bodengebundene Beobachtungen mit höherer Auflösung nicht direkt einbeziehen. Selbst das Hinzufügen einer Contrast Curve aus der LHO-Beobachtung verbessert das Ergebnis nicht signifikant ( $FPP = 0,987 \rightarrow 0,987$ ), da die dominanten Falsch-Positiv-Szenarien von bereits aufgelösten Sternen bei 7–20 Bogensekunden Abstand stammen, nicht von unaufgelösten Quellen innerhalb der Punktspreizfunktion.

Die entscheidende Information, dass der Transit *auf dem Zielstern* stattfindet, stammt aus den Bodenbeobachtungen von SaintEX und LHO Nowosibirsk, die den Zielstern räumlich von den Nachbarsternen trennen. Beide Instrumente messen eine Transittiefe von  $\sim 40$  ppt, die mit dem erwarteten Wert für einen Planeten auf dem M-Zwerg übereinstimmt. Diese Bodenbeobachtungen liegen ausserhalb des Anwendungsbereichs von TRICERATOPS.

Zusammenfassend ist die statistische Validierung mittels TRICERATOPS für TOI-7265.01 aufgrund der extremen Flussverdünnung in der TESS-Apertur nicht möglich. Die Validierung dieses Kandidaten stützt sich stattdessen auf die Konsistenz der Transitparameter über drei unabhängige Instrumente, die NEB-Analyse durch Alexey Garmash (2025), die chromatische Übereinstimmung der Transittiefen sowie die räumliche Auflösung der Bodenbeobachtungen und deren Posteriorübereinstimmung.

## 6.7 Charakterisierung von TOI 7265.01

Ohne die Masse ist die Klassifikation des Exoplaneten noch nicht bekannt, da keine Radialgeschwindigkeitsmethode angewendet wurde und die Masse durch die Transitmethode nicht berechnet werden kann, sodass man diese Aussage noch nicht exakt belegen kann. Deshalb wird die Masse-Radius-Korrelation benutzt (Müller et al., 2024), um zu approximieren, in welche Exoplanetenkategorie TOI-7265.01 fällt.

### 6.7.1 Die Masse-Radius-Relation für Riesenplaneten

Die Masse-Radius-Relation für Exoplaneten zeigt drei charakteristische Regimes (Müller & Helled, 2024): Für kleine, felsige Planeten gilt  $R \propto M^{0,27}$ , für Planeten mittlerer Masse mit Wasserstoff-Helium-Hüllen  $R \propto M^{0,67}$ , und für Gasriesen mit Massen oberhalb von  $\sim 127 M_{\oplus}$  ist der Radius nahezu massenunabhängig ( $R \propto M^{-0,06}$ ). Letzteres liegt am Einsetzen der Elektronenentartung im Planeteninnern, die bewirkt, dass zusätzliche Masse den Radius kaum noch vergrössert (Müller & Helled, 2024).

Für TOI-7265.01 b mit  $R_p = 10,5 \pm 0,6 R_{\oplus} \approx 0,94 R_{\text{Jup}}$  befindet sich das Objekt im jovianischen Regime. Hier führt die Inversion der Masse-Radius-Relation zu extrem grossen Unsicherheiten: Eine Änderung von wenigen Prozent im Radius kann die abgeleitete Masse um Grössenordnungen verschieben. Das probabilistische Modell *Forecaster* von Chen und Kipping (2017) liefert für diesen Radius eine Massenschätzung von  $M_p \sim 1,3 M_{\text{Jup}}$  mit einem  $1\sigma$ -Bereich von  $0,3\text{--}6 M_{\text{Jup}}$ . Diese grosse Streuung ist keine Schwäche des Modells, sondern eine intrinsische Eigenschaft des jovianischen Regimes.

## 6.7.2 Eingrenzung durch Vergleich mit ähnlichen Systemen

Eine engere Eingrenzung ergibt sich durch den Vergleich mit bekannten Riesenplaneten um M-Zwerg (Tabelle 6). Besonders relevant ist TOI-5205 b, ein jupitergrosser Planet ( $R = 1,03 R_{\text{Jup}}$ ,  $M = 1,08 M_{\text{Jup}}$ ) um einen M-Zwerg mit  $M_* = 0,39 M_{\odot}$  (Kanodia et al., 2023). Ebenfalls instruktiv ist TOI-3757 b, der masseärmste bekannte Gasriese um einen M-Zwerg ( $M = 0,27 M_{\text{Jup}}$ ) bei einem überraschend grossen Radius von  $1,07 R_{\text{Jup}}$  (Kanodia et al., 2023).

**Tabelle 6:** Bekannte Riesenplaneten um M-Zwerg im Vergleich zu TOI-7265.01. Die Massen wurden durch Radialgeschwindigkeitsmessungen bestimmt.

Planet	$R_p [R_{\text{Jup}}]$	$M_p [M_{\text{Jup}}]$	$P [\text{d}]$	$M_* [M_{\odot}]$
TOI-5205 b	1,03	1,08	1,72	0,39
TOI-5293 b	1,06	0,54	2,93	0,54
TOI-3757 b	1,07	0,27	3,44	0,57
TOI-4860 b	0,76	0,27	1,52	0,34
TOI-7265.01	0,94	?	4,17	0,49

Aus der Gleichgewichtstemperatur  $T_{\text{eq}} = 565 \text{ K}$  folgt, dass TOI-7265.01 ein *warmer Jupiter* ist, deutlich kühler als typische inflationierte Hot Jupiters. Aufblähung durch stellare Bestrahlung ist daher unwahrscheinlich, und die Dichte sollte in der Nähe typischer Gasriesen liegen ( $\rho \sim 0,5\text{--}1,5 \text{ g/cm}^3$ ). Unter der Annahme verschiedener mittlerer Dichten ergibt sich der in Tabelle 7 dargestellte Massenbereich.

**Tabelle 7:** Massenabschätzung für TOI-7265.01 unter verschiedenen Dichteannahmen.  $K$  bezeichnet die erwartete Radialgeschwindigkeits-Halbamplitude.

Dichteannahme	$\rho [\text{g/cm}^3]$	$M_p [M_{\text{Jup}}]$	$K [\text{m/s}]$
TOI-3757 b-analog	0,3	0,19	$\sim 38$
Saturn-analog	0,7	0,43	$\sim 88$
TOI-5205 b-analog	1,2	0,78	$\sim 158$
Jupiter-analog	1,3	0,83	$\sim 169$
Obere Grenze	2,0	1,25	$\sim 255$

## 6.7.3 Zusammenfassung der Massenabschätzung

Basierend auf der probabilistischen Masse-Radius-Relation von Chen und Kipping (2017), dem Vergleich mit ähnlichen Systemen und physikalischen Überlegungen zur Dichte ergibt sich für TOI-7265.01 eine geschätzte Masse von

$$M_p \approx 0.5 - 1.5 M_{\text{Jup}} \quad (160 - 480 M_{\oplus}), \quad ()$$

wobei die grosse Spannweite die intrinsische Entartung der Masse-Radius-Relation im jovianischen Regime widerspiegelt (Müller & Helled, 2024). Die erwartete RV-Halbamplitude von  $K \approx 100\text{--}300 \text{ m/s}$  liegt deutlich ueber der Nachweisgrenze moderner Spektrographen wie ESPRESSO ( $\sigma_K \sim 0,1 \text{ m/s}$ ), sodass eine präzise Massenbestimmung mit wenigen Beobachtungsstunden möglich wäre.

#### 6.7.4 Schlussfolgerung: TOI-7265.01 als echter Planetenkandidat

Die Ergebnisse des gemeinsamen Multi-Instrument-Fits erlauben eine robuste Schlussfolgerung über die Natur von TOI-7265.01. Alle drei unabhängigen Instrumente liefern Radiusverhältnisse, die innerhalb von  $2\sigma$  miteinander übereinstimmen: SaintEX ( $R_p/R_s = 0,2009$ ), LHO Nowosibirsk ( $R_p/R_s = 0,1988$ ) und TESS ( $R_p/R_s = 0,1800$ ). Die leichte Abweichung des TESS-Wertes nach unten ist auf die gröbere Pixelskala (21 Bogensekunden pro Pixel) und die daraus resultierende Flussverdünnung durch benachbarte Sterne zurückzuführen, die in den Bodenbeobachtungen räumlich aufgelöst werden konnten.

Das reduzierte Chi-Quadrat  $\chi^2_\nu$  liegt für alle drei Instrumente nahe 1 ( $\chi^2_\nu \approx 1,01$  bis  $1,23$ ), was auf eine gute Übereinstimmung zwischen Modell und Daten ohne Überanpassung hinweist (Sozzetti et al., 2007). Die Bodenbeobachtung in Nowosibirsk bestätigte unabhängig die Transitmitte und -tiefe. Die NEB-Analyse konnte 126 von 271 benachbarten Gaia-Sternen als Kontaminationsquellen ausschliessen; für die verbleibenden 48 nicht eindeutig ausgeschlossenen Sterne lieferte die räumliche Auflösung der SaintEX- und LHO-Beobachtungen den Nachweis, dass der Transit tatsächlich auf dem Zielstern TIC 275717567 stattfindet (Garmash, 2025).

Die statistische Validierung mittels TRICERATOPS ergab ein hohes FPP von 0,987, was jedoch nicht auf eine tatsächliche Falsch-Positiv-Natur hindeutet, sondern auf die extreme Flussverdünnung in der TESS-Apertur zurückzuführen ist: Der Zielstern trägt lediglich 4,3 % des Gesamtflusses bei. Dieses Ergebnis unterstreicht die Notwendigkeit bodengebundener Nachbeobachtungen mit höherer räumlicher Auflösung.

Aufgrund der übereinstimmenden Transittiefen aus drei unabhängigen Instrumenten, der Konsistenz aller Transitparameter und der Abwesenheit identifizierbarer Falsch-Positiv-Szenarien kann TOI-7265.01 als *robuster Planetenkandidat* klassifiziert werden. Tabelle 8 fasst die abgeleiteten Planetenparameter zusammen. Es handelt sich mit überragend grosser Wahrscheinlichkeit um einen **echten und neuentdeckten Exoplaneten**.

Mit einem Radius von  $R_p = 0,938 \pm 0,023 R_{\text{Jup}}$  und einer geschätzten Masse von  $0,5 - 1,5 M_{\text{Jup}}$  fällt TOI-7265.01 in die Kategorie der Gasriesen nach der Klassifikation von Chen und Kipping (2017). Gasriesen um M-Zwerg sind ausserordentlich selten: Ihre Vorkommenshäufigkeit beträgt weniger als 1 % (Kanodia et al., 2023). Die erwartete Radialgeschwindigkeits-Halbamplitude von  $K \approx 100 - 300$  m/s liegt deutlich über der Nachweisgrenze moderner Spektrographen wie ESPRESSO, sodass eine präzise Massenbestimmung und damit die endgültige Bestätigung als Planet mit wenigen Beobachtungsstunden möglich wäre.

**Tabelle 8:** Zusammenfassung der abgeleiteten Parameter von TOI-7265.01.

Parameter	Wert	Quelle
<i>Orbitalparameter</i>		
Periode $P$	$4,1723 \pm 0,0001$ d	Joint Fit
Transitmitte $T_0$ (BJD)	$2460564,7804 \pm 0,0005$	Joint Fit
Inklination $i$	$\sim 89$	abgeleitet
Exzentrizität $e$	0 (fixiert)	Annahme
<i>Transitparameter</i>		
$R_p/R_s$ (SaintEX)	$0,2009 \pm 0,0055$	Joint Fit
$R_p/R_s$ (LHO)	$0,1988 \pm 0,0078$	Joint Fit
$R_p/R_s$ (TESS)	$0,1800 \pm 0,0042$	Joint Fit
Impact-Parameter $b$	$0,30 \pm 0,10$	Joint Fit
Transitdauer $T_{14}$	$\sim 101$ min	abgeleitet
<i>Physikalische Parameter</i>		
Planetenradius $R_p$	$10,52 \pm 0,25 R_{\oplus}$	Joint Fit
Planetenradius $R_p$	$0,938 \pm 0,023 R_{\text{Jup}}$	Joint Fit
Geschätzte Masse $M_p$	$0,5 - 1,5 M_{\text{Jup}}$	M-R-Relation
Gleichgewichtstemperatur $T_{\text{eq}}$	$564,9 \pm 6,0$ K	abgeleitet
<i>Stellare Parameter (TIC 275717567)</i>		
Sternmasse $M_{\star}$	$0,486 \pm 0,021 M_{\odot}$	ExoFOP
Sternradius $R_{\star}$	$0,488 \pm 0,015 R_{\odot}$	ExoFOP
Effektivtemperatur $T_{\text{eff}}$	$3392 \pm 200$ K	ExoFOP
Sterndichte $\rho_{\star}$	$5,891 \text{ g/cm}^3$	Joint Fit
Spektraltyp	M3V	ExoFOP

## 7 Reflexion

Das in dieser Arbeit gewählte Vorgehen hat sich als geeignet erwiesen, um ein differenziertes Parameterbild für den Exoplanetenkandidaten TOI-7265.01 zu erhalten. Die zentrale methodische Stärke liegt im gemeinsamen Multi-Instrument-Fit, der die Lichtkurven von SaintEX, dem Lyceum 130 Observatorium (LHO) in Nowosibirsk und TESS simultan mit einem 19-dimensionalen Parametervektor modelliert. Die Reparametrisierungen nach Espinoza (2018) und Kipping (2013) haben entscheidend dazu beigetragen, dass die MCMC-Ketten effizient konvergierten und die Posteriorverteilungen annähernd gauss'sche Formen annahmen, was eine zuverlässige Fehlerquantifizierung ermöglicht.

Die Konsistenz der Ergebnisse über drei unabhängige Instrumente ist das stärkste Argument für die astrophysikalische Natur des Signals. Die Radiusverhältnisse  $R_p/R_s$  stimmen innerhalb von  $2\sigma$  überein (SaintEX:  $0,201 \pm 0,006$ , LHO:  $0,199 \pm 0,008$ , TESS:  $0,180 \pm 0,004$ ), und die reduzierten  $\chi^2_{\nu}$ -Werte liegen für alle drei Datensätze nahe 1. Die leicht tiefere TESS-Messung ist physikalisch erklärbar durch die Flussverdünnung in der grossen TESS-Apertur (21 Bogensekunden pro Pixel), in der der Zielstern nur 4,3% des Gesamtflusses beiträgt. Gerade dieser Umstand unterstreicht den Wert der Bodenbeobachtungen: Erst die räumliche Auflösung von SaintEX und LHO ermöglichte es, den Transit eindeutig dem Zielstern TIC 275717567 zuzuordnen.

Im Bereich der statistischen Validierung wurden sowohl TRICERATOPS (Giacalone et al., 2021) als auch WATSON-Net (Dévora-Pajares et al., 2025) evaluiert. TRICERATOPS ergab ein FPP von 0,987, was jedoch nicht auf eine Falsch-Positiv-Natur hindeutet, sondern auf die extreme Flussverdünnung im TESS-Gesichtsfeld zurückzuführen ist. WATSON-Net konnte nicht vollständig ausgeführt werden, da für TIC 275717567 keine SPOC-Pipeline-Datenprodukte vorliegen, die das neuronale Netzwerk als Eingabe benötigt. Beide Ergebnisse zeigen, dass rein TESS-basierte Validierungswerkzeuge für Ziele in dicht besiedelten Sternfeldern mit dominanten Nachbarsternen an ihre Grenzen stossen. Die Validierung stützt sich daher auf die Multi-Instrument-Konsistenz, die NEB-Analyse (126 von 271 Nachbarsternen ausgeschlossen) und die räumliche Bestätigung des Transits durch die Bodenbeobachtungen.

Eine wesentliche Einschränkung bleibt das Fehlen einer Radialgeschwindigkeitsmessung. Ohne gemessene Masse kann nicht ausgeschlossen werden, dass das Objekt ein Brauner Zwerg oder ein sehr kleiner Stern ist. Die Masse-Radius-Relation von Chen und Kipping (2017) liefert eine Abschätzung von  $M_p \approx 0,5 - 1,5 M_{\text{Jup}}$ , jedoch mit intrinsisch grosser Unsicherheit, da der Radius von Gasriesen nahezu massenunabhängig ist (Müller & Helled, 2024). Die erwartete RV-Halbamplitude von  $K \approx 100 - 300$  m/s liegt deutlich über der Nachweisgrenze moderner Spektrographen, sodass eine endgültige Bestätigung mit wenigen Beobachtungsstunden möglich wäre.

Weitere Kritikpunkte betreffen die Behandlung systematischer Fehlerquellen. Red-Noise-Effekte wurden durch die Einführung eines instrumentenspezifischen Jitter-Terms ( $\ln \sigma_w$ ) im Modell berücksichtigt, jedoch nicht durch einen dedizierten Gaussian-Prozess-Regressionsansatz. Die Limb-Darkening-Koeffizienten wurden als freie Parameter pro Datensatz gefittet, was die Flexibilität erhöht, aber bei den relativ verrauschten NOVO-Daten (62 Datenpunkte) zu grösseren Unsicherheiten führt.

Auf Basis dieser Maturaarbeit wird in Zusammenarbeit mit Prof. Dr. Brice-Olivier Demory und Francis Zong Lang (Universität Bern) ein wissenschaftliches Paper entstehen, in welchem die hiesigen Resultate mit weiteren Methoden bestätigt und in einen astrophysikalischen Kontext eingeordnet werden. Gasriesen um M-Zwerges gehören zu den seltensten Planetentypen in der Galaxis (Vorkommenshäufigkeit  $< 1\%$ ), und TOI-7265.01 ist damit ein wertvoller Prüfstein für die Planetenentstehungstheorie, die vorhersagt, dass M-Zwerges aufgrund ihrer geringen Masse kaum Gasriesen bilden können.

Rückblickend lässt sich festhalten, dass die Methodik den Ansprüchen an Transparenz und Nachvollziehbarkeit genügt und durch den Multi-Instrument-Ansatz eine Robustheit erreicht, die über eine rein TESS-basierte Analyse hinausgeht. Die Grenzen der automatisierten Validierungswerkzeuge wurden klar aufgezeigt, und es wurde demonstriert, dass bodengebundene Follow-up-Beobachtungen für Ziele in überfüllten Sternfeldern unverzichtbar bleiben.

## 8 Danksagung und Widmung

Mein besonderer Dank gilt Prof. Dr. Brice-Olivier Demory und Francis Zong Lang von der Universität Bern, die mir mit ihrer externen Hilfe, ihrem Fachwissen und ihrer aufrichtigen Unterstützung zur Seite standen. Ohne ihren Einsatz hätte diese Arbeit in dieser Form nicht realisiert werden können. Ebenso möchte ich der gesamten Forschungsgruppe des Saint-Ex-Observatoriums danken, welche die Beobachtungen für meine Maturaarbeit in Mexiko durchgeführt hat. Grossen Dank verdient ausserdem Dr. Erik Meier Valdés, welcher meine Arbeit im Rahmen des nationalen Finale des Schweizer Jugend Forscht Wettbewerbs betreute. Ich möchte mich ausserdem speziell bei Dr. Mathilde Timmermans bedanken, da ich ohne ihre Hilfe nie eine Spezialistin zur ExoFOP-Datenbank an meiner Seite gehabt hätte. Mein Dank richtet sich auch an Dr. Valentin Bickel, der mir den Kontakt zu Prof. Dr. Demory und Herrn Zong vermittelt und damit die Zusammenarbeit erst ermöglicht hat. Natürlich verdienen auch meine betreuende Lehrperson Michael Franck und der Korreferent Dr. Markus Ninck einen Dank, da ich ohne die beiden die Arbeit nie an meiner Kantonsschule realisieren könnte. Ohne die Türen, die mir durch die Begabtenförderung an der Kantonsschule Zofingen geöffnet wurden, wäre ich vermutlich nie in Kontakt mit all den genannten Personen getreten. Deshalb spreche ich auch Herrn Alexander Fend, dem Koordinator der Begabtenförderung und der Schulleitung der Kantonsschule Zofingen einen besonderen Dank aus, da ohne die Unterstützung dieser Personen ich nie in Kontakt mit meiner externen Betreuung getreten wäre. Die grosszügige Unterstützung aller Parteien war für das Gelingen dieser Maturaarbeit entscheidend.

Ausserdem möchte ich diese Arbeit und den darin charakterisierten Warm Jupiter TOI-7265.01 meinem verstorbenen Grossvater Milivoje Mihajlović widmen, welcher stets ein geduldiger und interessierter Zuhörer war und mich trotz seiner nur vier Jahren dauernden Grundschulbildung immer bestärkt und dazu ermuntert hat, weiterzumachen. Ihm widme ich diese Arbeit, die ohne seine Fürsorge und Ermutigung so nie möglich gewesen wäre. Ich hoffe er sieht stolz auf mich und den ungefähr 470 Lichtjahre entfernten Exoplaneten herab.

## 9 Referenzen

- Anaconda Software Distribution. (2025). *Anaconda*. <https://anaconda.com>
- Bonnarel, F., Fernique, P., Bienaymé, O., Egret, D., Genova, F., Louys, M., Ochsenbein, F., Wenger, M., & Bartlett, J. G. (2000). The ALADIN interactive sky atlas. A reference tool for identification of astronomical sources.
- Boyce, G., & Conti, D. (2016). *The AstrolmageJ Cookbook*.
- Chen, J., & Kipping, D. M. (2017). Probabilistic Forecasting of the Masses and Radii of Other Worlds. *The Astrophysical Journal*, 834(17).
- Collins, K. A., Kielkopf, J. F., Stassun, K. G., & Hessman, F. V. (2017). AstrolmageJ: Image Processing and Photometric Extraction for Ultra-Precise Astronomical Light Curves (Expanded Edition). *The Astronomical Journal*, 153.
- Demory, B.-O. (2024). Discovering transiting exoplanets [unveröffentlichtes Übungsskript für einen Laborkurs]. Universität Bern.
- Dévora-Pajares, M., Pozuelos, F., Suárez, J., González-Penedo, M., & Dafonte, C. (2025). WATSON-Net: Vetting, Validation, and Analysis of Transits from Space Observations with Neural Networks. *Astronomy & Astrophysics*.
- Espinoza, N. (2018). Efficient Joint Sampling of Impact Parameters and Transit Depths in Transiting Exoplanet Light Curves. *Research Notes of the AAS*, 2(209).
- Fiveable Content Team. (2025). *Exoplanet classification and diversity*. <https://fiveable.me/astrobiology/unit-9/exoplanet-classification-diversity/study-guide/OzXOqWOQb6VscJi0>
- Ford, E. B. (2005). Quantifying the Uncertainty in the Orbits of Extrasolar Planets. *The Astronomical Journal*, 129(3).
- Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. (2013). emcee: The MCMC Hammer. *Publications of the Astronomical Society of the Pacific*, 125.
- Garcia, L. J., Foreman-Mackey, D., Murray, C. A., Timmermans, M., Delrez, L., Roy, P.-A., & Gillon, M. (2024). nuance: efficient detection of planets transiting active stars. *Monthly Notices of the Royal Astronomical Society*, 528, 4202–4215.
- Garmash, A. (2025). ExoFOP-TESS Beobachtungsbericht für TOI-7265.01 [Lyceum 130 Observatory, Nowosibirsk]. *ExoFOP-TESS*.
- Giacalone, S., Dressing, C. D., Jensen, E. L. N., Collins, K. A., Ricker, G. R., Vanderspek, R., Seager, S., Winn, J. N., Jenkins, J. M., Barclay, T., Barkaoui, K., Cadieux, C., Charbonneau, D., Collins, K. I., Conti, D. M., Doyon, R., Evans, P., Ghachoui, M., Gillon, M., . . . Waite, I. A. (2021). Vetting of 384 TESS Objects of Interest with TRICERATOPS and Statistical Validation of 12 Planet Candidates. *The Astronomical Journal*, 161.
- Gregory, P. C. (2007). A Bayesian Kepler periodogram detects a second planet in HD 208487. *Monthly Notices of the Royal Astronomical Society*, 374, 1321–1333.
- Guerrero, N. M., Seager, S., Huang, C. X., Vanderburg, A., Soto, A. G., Mireles, I., Hesse, K., Fong, W., Glidden, A., Shporer, A., Latham, D. W., Collins, K. A., Quinn, S. N., Burt, J., Dragomir, D., Crossfield, I., Vanderspek, R., Fausnaugh, M., Burke, C. J., . . . Winn, J. N. (2021). The TESS Objects of Interest Catalog from the Prime Mission. *The Astrophysical Journal Supplement Series*.
- Hastings, W. K. (1969). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57, 97.
- Jensen, E. (2013). Tapir: A web interface for transit/eclipse observability. *Astrophysics Source Code Library*.
- Kanodia, S., Mahadevan, S., Libby-Roberts, J., Stefansson, G., Cañas, C. I., Lin, A. S. J., Gupta, A. F., Han, T., Kobulnicky, H. A., Larsen, A., Robertson, P., Ninan, J. P., Bender, C. F., Cochran, W. D., Endl, M., Ford, E. B., Halverson, S., Hearty, F., Monson, A., . . . Schwab, C. (2023). TOI-5205b: A Short-period Jupiter-sized Planet Transiting a Mid-M

- Dwarf. *The Astronomical Journal*, 165(3), 120. <https://doi.org/10.3847/1538-3881/acabce>
- Kipping, D. M. (2013). Efficient, uninformative sampling of limb darkening coefficients for two-parameter laws. *Monthly Notices of the Royal Astronomical Society*, 435, 2152–2162.
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows.
- Kopparapu, R. K., Ramirez, R., Kasting, J. F., Eymet, V., Robinson, T. D., Mahadevan, S., Terrien, R. C., Domagal-Goldman, S., Meadows, V., & Deshpande, R. (2013). HABITABLE ZONES AROUND MAIN-SEQUENCE STARS: NEW ESTIMATES. *The Astrophysical Journal*, 765.
- Kreidberg, L. (2015). batman: BAsic Transit Model cAlculation in Python. *Publications of the Astronomical Society of the Pacific*, 127.
- Loeb, A., & Gaudi, B. S. (2018). PERIODIC FLUX VARIABILITY OF STARS DUE TO THE REFLEX DOPPLER EFFECT INDUCED BY PLANETARY COMPANIONS. *The Astrophysical Journal*.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., & Teller, A. H. (1953). Equation of State Calculations by Fast Computing Machines and Edward Teller. *The Journal of Chemical Physics*, 21(6).
- Müller, S., Baron, J., Helled, R., Bouchy, F., & Parc, L. (2024). Revising the M–R Relations of Exoplanets Using the PlanetS Catalog. *Astronomy & Astrophysics*, 686.
- Müller, S., & Helled, R. (2024). The mass-radius relation of exoplanets, revisited. *Astronomy & Astrophysics*, 686, A296.
- NASA. (2025a). *Exoplanets*. Zugriff 20. September 2025 unter <https://science.nasa.gov/exoplanets>
- NASA. (2025b). *What is an Exoplanet?* <https://science.nasa.gov/exoplanets/planet-types>
- NASA und California Institute of Technology. (2025). *ExoFOP*. Zugriff 20. Oktober 2025 unter <https://www.ipac.caltech.edu/project/exofop>
- Practical Astrophotography Staff. (2015). *A Brief Guide to Calibration Frames: Bias, Dark, Flats and Dark Flats*. Zugriff 20. Oktober 2025 unter <https://practicalastrophotography.com/a-brief-guide-to-calibration-frames>
- Ricker, G. R., Winn, J. N., Vanderspek, R., Latham, D. W., Bakos, G. A., Bean, J. L., Berta-Thompson, Z. K., Brown, T. M., Buchhave, L., Butler, N. R., Butler, R. P., Chaplin, W. J., Charbonneau, D., Christensen-Dalsgaard, J., Clampin, M., Deming, D., Doty, J., Lee, N. D., Dressing, C., . . . Villaseñor, J. (2014). The Transiting Exoplanet Survey Satellite. *Journal of Astronomical Telescopes, Instruments, and Systems*.
- Schlecker, M., Burn, R., Sabotta, S., Seifert, A., Henning, T., Emsenhuber, A., Mordasini, C., Reffert, S., Shan, Y., & Klahr, H. (2022). RV-detected planets around M dwarfs: Challenges for core accretion models.
- Seager, S., & Mallén-Ornellas, G. (2003). A Unique Solution of Planet and Star Parameters from an Extrasolar Planet Transit Light Curve. *The Astrophysical Journal*, 585, 1038–1055.
- Singal, R., & Rana, R. K. (2015). Chi-square test and its application in hypothesis testing. *Journal of the Practice of Cardiovascular Sciences*, 1.
- Sozzetti, A., Torres, G., Charbonneau, D., Latham, D. W., Holman, M. J., Winn, J. N., Laird, D. W., & O'Donovan, E. J. (2007). Improving Stellar and Planetary Parameters of Transiting Planet Systems: The Case of TrES-2. *The Astrophysical Journal*, 664, 1190–1198.
- The Planetary Society. (2025). *The Different Kinds of Exoplanets You Meet in the Milky Way*. <https://www.planetary.org/articles/the-different-kinds-of-exoplanets-you-meet-in-the-milky-way>

- Tuchow, N. W., Ford, E. B., Papamarkou, T., & Lindo, A. (2019). The efficiency of geometric samplers for exoplanet transit timing variation models. *Monthly Notices of the Royal Astronomical Society*, 484, 3772–3784.
- University of Exeter. (2025). *Astronomical bodies: exoplanets*. <https://www.exeter.ac.uk/research/groups/physics/astrophysics/outreach/learningresources/landerresources/exoplanets>
- Winn, J. N. (2014). *Transits and Occultations*.

## 10 Abbildungsverzeichnis

1	<i>Schematische Darstellung eines Hot Jupiters. (Bildquelle: ESO/M. Kornmesser)</i>	1
2	<i>Schematische Darstellung einer Lichtkurve beim Transit eines Exoplaneten vor seinem Mutterstern. (Bildquelle: NASA)</i>	9
3	<i>Illustration der Radialgeschwindigkeitsmethode: Darstellung der Bewegung von Stern und Planet sowie der Dopplerverschiebungen. (Bildquelle: SUW-Grafik; Erde: Terraforma)</i>	11
4	<i>Direkte Bildgebung eines extrasolaren Planeten. Die Aufnahme zeigt, wie ein Exoplanet im direkten Bild von seinem Mutterstern getrennt detektiert werden kann. (Bildquelle: Jason Wang et al., 2008)</i>	12
5	<i>Das Saint-Ex-Observatorium in San Pedro Mártir, Mexiko. (Bildquelle: Institute of Astronomy, UNAM / E. Cadena)</i>	13
6	<i>Foto des TESS-Satelliten im Labor. (Orbital ATK / NASA)</i>	13
7	Output des Swarthmore College Transit Finders für TOI 7265.01.	22
8	Eine Rohaufnahme des Sternfelds um TOI 7265.01.	22
9	Verunreinigtes Rohbild mit heller Störspur, vermutlich durch einen Satelliten- oder Meteoritenflug verursacht.	23
10	Das Master-Dark-Frame für die Beobachtung von TOI 7265.01.	24
11	Das Master-Flat-Frame für die Beobachtung von TOI 7265.01.	24
12	Das Master-Bias-Frame für die Beobachtung von TOI 7265.01.	25
13	<i>Das Sternfeld um TOI 7265.01 nach Dark-, Bias- und Flat-Kalibrierung.</i>	25
14	Die um 90° gedrehte Eigenbeobachtung mit rotmarkiertem Zielstern.	26
15	Die aus Aladin stammende Identifikationshilfe.	26
16	<i>Die eingezeichneten Aperturen des Zielsterns (T1) und der Referenzsterne (C#).</i>	27
17	<i>Unbereinigte Lichtkurve des Zielsterns während des Transits.</i>	28
18	<i>Durch Luftmasse und FWHM trendbereinigte Lichtkurve.</i>	28
19	Transitmodell und Residuenplot von TOI 7265.01	41
20	Corner-Plot der Parameter für die Saint-Ex-Transitdaten	42
21	Corner-Plot der Parameter für die TESS-Transitdaten.	44
22	<i>Corner-Plot der Posteriores des gemeinsamen Multi-Instrument-Fits. Die neun dargestellten Parameter zeigen annähernd gauss'sche, dekorrelierte Verteilungen.</i>	46
23	<i>Das Transitmodell (rote Linie) überlagert auf die Beobachtungsdaten aller drei Instrumente. Die Residuen sind im unteren Panel dargestellt.</i>	47
24	<i>Phasengefaltete Lichtkurve aller drei Datensätze, zentriert auf die Transitmitte (<math>\phi = 0</math>). Die Datenpunkte sind farblich nach Instrument kodiert, die rote Linie zeigt das Medianmodell.</i>	47

25	<i>Kerndichteschätzung (KDE) der Posteriores-Verteilung des Radiusverhältnisses <math>R_p/R_s</math>.</i> . . . . .	48
26	TRICERATOPS-Szenario-Wahrscheinlichkeiten für TOI-7265.01. Der Zielstern TIC 275717567 erhält nur 1,3% Planetenwahrscheinlichkeit, da TRICERATOPS den Grossteil der Wahrscheinlichkeit den helleren Nachbarsternen zuordnet, die innerhalb der TESS-Apertur liegen. . . . .	49
27	Flussanteile in der TESS-Apertur (links) im Vergleich zur SaintEX-Apertur (rechts). In der TESS-Apertur trägt der Zielstern nur 4,3% des Gesamtflusses bei, was TRICERATOPS dazu veranlasst, die Wahrscheinlichkeit den helleren Nachbarsternen zuzuordnen. . . . .	50

## 11 Tabellenverzeichnis

1	Vergleich der vorhergesagten und gemessenen Transitparameter (LHO, 2025.10.15)	17
2	Parametervektor des gemeinsamen Transit-Fits (19 Dimensionen)	39
3	Globale Transitparameter des gemeinsamen Multi-Instrument-Fits für TOI-7265.01	45
4	Instrumentenspezifische Radiusverhältnisse und reduziertes Chi-Quadrat ( $\chi^2_\nu$ )	45
5	Dominante Nachbarsterne innerhalb der TESS-Apertur von TOI-7265 und ihre TRICERATOPS-Wahrscheinlichkeiten. Der Zielstern trägt nur 4,3% des Gesamtflusses bei.	50
6	Bekannte Riesenplaneten um M-Zwerg im Vergleich zu TOI-7265.01. Die Massen wurden durch Radialgeschwindigkeitsmessungen bestimmt.	52
7	Massenabschätzung für TOI-7265.01 unter verschiedenen Dichteannahmen. $K$ bezeichnet die erwartete Radialgeschwindigkeits-Halbamplitude.	52
8	Zusammenfassung der abgeleiteten Parameter von TOI-7265.01.	54

## 12 Formelverzeichnis

### Formel 1: Planetenradius

$$R_p = R_s \times \sqrt{\delta}$$

Erklärungen:  $R_p$ : Radius des Exoplaneten, bestimmt über den Anteil des vom Stern abgeschatteten Lichts.  $R_s$ : Radius des Sterns, um den der Planet kreist.  $\delta$ : Transittiefe, der relative Helligkeitsabfall während des Transits.

### Formel 2: Umlaufperiode

$$P = \frac{2\pi a^{3/2}}{\sqrt{GM_\star}}$$

Erklärungen:  $P$ : Zeit, die der Planet für eine vollständige Umrundung des Sterns benötigt (Periode).  $a$ : Grosse Halbachse der elliptischen Umlaufbahn.  $G$ : Gravitationskonstante, beschreibt die Stärke der Gravitation.  $M_\star$ : Masse des Sterns.

### Formel 3: Transittiefe

$$\delta = \left(\frac{R_p}{R_s}\right)^2$$

Erklärungen:  $\delta$ : Der Anteil des gestreuten Sternlichts, welcher durch den Planeten verdeckt wird.  $R_p, R_s$ : Siehe oben.

### Formel 4: Transitdauer

$$T_{14} = \frac{PR_s}{\pi a} \sqrt{\left(1 + \frac{R_p}{R_s}\right)^2 - b^2}$$

Erklärungen:  $T_{14}$ : Dauer des gesamten Transits von Eintritt bis Austritt.  $P, R_s, a, R_p$ : Wie oben.  $b$ : Stossparameter, der den kürzesten Abstand des Planetenpfads zum Sternzentrum normiert durch Sternradius beschreibt.

### Formel 5: Mittlerer Transitzeitpunkt

$$t_0 = T_c \quad (\text{bestimmt aus der Lichtkurve})$$

Erklärungen:  $t_0$ : Zeitpunkt, an dem der Planet die Zentralsternscheibe überdeckt.  $T_c$ : Gemessener zentraler Zeitwert aus der Transitlichtkurve.

### Formel 6: Mittlere Sterndichte

$$\rho_{\star} = \frac{3\pi}{GP^2} \left( \frac{a}{R_s} \right)^3$$

Erklärungen:  $\rho_{\star}$ : Dichte des Sterns, abgeleitet aus orbitalen Parametern.  $G$ : Gravitationskonstante.  $P$ : Umlaufperiode.  $a$ : Grosse Halbachse.  $R_s$ : Sternradius.

### Formel 7: Verhältnis Planetenradius zum Sternradius

$$\frac{R_p}{R_s} = \sqrt{\delta}$$

Erklärungen:  $R_p, R_s, \delta$ : Wie oben.

### Formel 8: Limb-Darkening Koeffizienten

Erklärungen:  $u_1, u_2$ : Parameter zur Beschreibung der Abnahme der Sternhelligkeit vom Zentrum zum Rand, experimentell durch Beobachtung der Lichtkurve bestimmt.

### Formel 9: Espinoza (2018) Reparametrisierung

Fall 1 ( $r_1 > A_r$ ), mit  $A_r = (p_U - p_L)/(2 + p_L + p_U)$ :

$$b = (1 + p_L) \left( 1 + \frac{r_1 - 1}{1 - A_r} \right), \quad p = (1 - r_2) p_L + r_2 p_U$$

Fall 2 ( $r_1 \leq A_r$ ), mit  $q = r_1/A_r$ :

$$b = (1 + p_L) \sqrt{q} r_2, \quad p = p_U + (p_L - p_U) \sqrt{q} (1 - r_2)$$

Erklärungen:  $r_1, r_2$ : Hilfsparameter, gleichverteilt auf  $[0, 1]$ .  $b$ : Stossparameter.  $p = R_p/R_s$ : Radiusverhältnis Planet zu Stern.  $p_L, p_U$ : Untere und obere Grenze für das Radiusverhältnis.  $A_r$ : Skalierungsparameter der Transformation.

### Formel 10: Kipping (2013) Limb-Darkening Transformation

$$u_1 = 2\sqrt{q_1} q_2, \quad u_2 = \sqrt{q_1} (1 - 2q_2)$$

Erklärungen:  $q_1, q_2$ : Hilfsparameter, gleichverteilt auf  $[0, 1]$ ; beschreiben das physikalisch zulässige Dreieck in  $(u_1, u_2)$ .  $u_1, u_2$ : Quadratische Limb-Darkening-Koeffizienten (physikalisch).

### Formel 11: Logarithmierte Likelihood-Funktion

$$\ln \mathcal{L}(\theta) = -\frac{1}{2} \sum_i \left[ \frac{(y_i - m_i)^2}{\sigma_i^2 + \sigma_w^2} + \ln(2\pi(\sigma_i^2 + \sigma_w^2)) \right]$$

Erklärungen:  $y_i$ : Gemessener Flusswert des  $i$ -ten Datenpunkts.  $m_i$ : Vorhergesagter Modellfluss für Parameter  $\theta$ .  $\sigma_i$ : Formaler Messfehler des  $i$ -ten Datenpunkts.  $\sigma_w$ : Jitter-Parameter (zusätzliches weisses Rauschen).

### Formel 12: Bayes-Theorem

$$P(\theta | D) = \frac{P(D | \theta) P(\theta)}{P(D)}$$

Erklärungen:  $P(\theta | D)$ : Posteriore Wahrscheinlichkeit der Parameter  $\theta$  gegeben die Daten  $D$ .  $P(D | \theta)$ : Likelihood – Wahrscheinlichkeit der Daten gegeben die Parameter.  $P(\theta)$ : Prior – Vorwissen über die Parameter vor der Beobachtung.  $P(D)$ : Evidenz (Normierungskonstante).

### Formel 13: Gleichgewichtstemperatur

$$T_{\text{eq}} = T_{\star} \sqrt{\frac{R_{\star}}{2a}}$$

Erklärungen:  $T_{\text{eq}}$ : Gleichgewichtstemperatur des Planeten (Annahme: Albedo = 0, gleichmäßige Wärmeverteilung).  $T_{\star}$ : Effektive Temperatur des Sterns.  $R_{\star}$ : Radius des Sterns.  $a$ : Grosse Halbachse der Planetenbahn.

### Formel 14: Einstrahlungsfluss

$$S = \frac{L_{\star}}{a^2}$$

Erklärungen:  $S$ : Einstrahlungsfluss des Planeten in Einheiten des Erdeinflusses  $S_{\oplus}$ .  $L_{\star}$ : Leuchtkraft des Sterns.  $a$ : Grosse Halbachse der Planetenbahn.

## 13 Anhang

Im folgenden Kapitel liegen alle Anhänge dieser wissenschaftlichen Arbeit.

### 13.1 Python-Quellcode des Joint-Fit-Notebooks

Der folgende Abschnitt enthält den vollständigen Python-Quellcode des Jupyter-Notebooks, das den gemeinsamen Multi-Instrument-Transit-Fit für TOI-7265.01 durchführt. Das Notebook verwendet emcee (Foreman-Mackey et al., 2013) als MCMC-Sampler und batman (Kreidberg, 2015) für die analytische Transitmodellierung.

#### Listing 10: *Importe und Thread-Konfiguration*

```
# Cell 1: Imports
import os
os.environ["OMP_NUM_THREADS"] = "1"
os.environ["MKL_NUM_THREADS"] = "1"
os.environ["NUMEXPR_NUM_THREADS"] = "1"
import numpy as np
import batman
import scipy.stats as stats
import emcee
import multiprocessing as mp
from multiprocessing import get_context
from collections import OrderedDict
import matplotlib.pyplot as plt
import corner
import pandas as pd
from IPython.display import display, HTML
```

#### Listing 11: *Das Modul transitworker.py*

```
%%writefile transitworker.py
"""
transitworker.py - TOI-7265.01 joint multi-instrument transit fit (v9)
=====
Changes from v8:
- log_rho sigma = 0.08 (was 0.12): tighter density prior
- b prior sigma = 0.10 (was 0.15): narrower impact parameter prior
- PU = 0.28 (was 0.30): tighter Rp/R* upper bound
- t0 sigma = 2*DUR/24 (was 3*DUR/24): tighter mid-transit prior
- log_sw upper = ln(50) (was ln(100)): cap jitter
- nsteps = 20000, burn-in = 65%
Parameter vector (N_datasets), TOTAL = 3 + 5*N:
0 t0 Gaussian(N(TO_REF, 2*DUR/24))
1 log_rho_star Gaussian(N(log(rho_ref), 0.08))
2 r1_global Uniform[0, 1] -> b (+ Gaussian prior on b)
3..3+N-1 r2_i Uniform[0, 1] -> p_i per dataset
3+N..3+2N-1 q1_i Uniform[0, 1] -> u1_i (Kipping 2013)
3+2N..3+3N-1 q2_i Uniform[0, 1] -> u2_i
3+3N..3+4N-1 mean_i Gaussian(N(0, 10*median_err))
3+4N..3+5N-1 log_sw_i Uniform(log[1e-4, 50]) ppt
References:
- Espinoza 2018 RNAAS 2 209
- Kipping 2013 MNRAS 435 2152
```

```

Espinoza+2019(juliet)MNRAS4902262
Sozzetti+2007ApJ6641190
"""
import numpy as np
import scipy.stats as stats
import batman
DATASETS = None
PERIOD_GUESS = None
TO_REF = None
DUR_GUESS = None
DEPTH_GUESS = None
RHO_STAR_GUESS = None
def init_worker(shared_datasets, period_guess, t0_ref, dur_guess,
                depth_guess, rho_star_guess):
    global DATASETS, PERIOD_GUESS, TO_REF, DUR_GUESS, DEPTH_GUESS,
           RHO_STAR_GUESS
    DATASETS = shared_datasets
    PERIOD_GUESS = period_guess
    TO_REF = t0_ref
    DUR_GUESS = dur_guess
    DEPTH_GUESS = depth_guess
    RHO_STAR_GUESS = rho_star_guess
#
-----

# Physical helpers
#
-----

def _a_over_rs(rho_cgs, period_days):
    """Kepler III: a/R from stellar density and period."""
    G = 6.67430e-11
    rho_SI = rho_cgs * 1000.0
    P_sec = period_days * 86400.0
    return ((G * rho_SI * P_sec**2) / (3.0 * np.pi)) ** (1.0 / 3.0)
#
-----

# Espinoza (2018): (r1, r2) -> (b, p)
#
-----

_PL = 0.03 # min Rp/R*
_PU = 0.28 # max Rp/R* (tightened from 0.30)
def _r1r2_to_bp(r1, r2, pl=_PL, pu=_PU):
    """Espinoza(2018) eq. 1-2. Returns (b, p) given r1, r2 in [0, 1]."""
    Ar = (pu - pl) / (2.0 + pl + pu)
    if r1 > Ar:
        b = (1.0 + pl) * (1.0 + (r1 - 1.0) / (1.0 - Ar))
        p = (1.0 - r2) * pl + r2 * pu
    else:
        q = r1 / Ar
        b = (1.0 + pl) * np.sqrt(q) * r2
        p = pu + (pl - pu) * np.sqrt(q) * (1.0 - r2)
    return b, p
#
-----

```

```

# Kipping (2013): (q1, q2) -> (u1, u2)
#
-----

def _q1q2_to_u1u2(q1, q2):
    """Kipping (2013) eq. 15-16."""
    u1 = 2.0 * np.sqrt(q1) * q2
    u2 = np.sqrt(q1) * (1.0 - 2.0 * q2)
    return u1, u2
#
-----

# Batman transit model with dilution
#
-----

def transit_model_diluted(b, p, t0, rho_star, D, x, u1, u2):
    """Returns flux in ppt. D=dilution: obs=D*batman+(1-D)."""
    period = PERIOD_GUESS
    if not np.isfinite([b, p, t0, rho_star, D, u1, u2]).all():
        return np.full_like(x, np.nan)
    if rho_star <= 0 or b < 0 or p <= 0 or not (0.0 < D <= 1.0):
        return np.full_like(x, np.nan)
    a_rs = _a_over_rs(rho_star, period)
    if not np.isfinite(a_rs) or a_rs <= 0 or b >= a_rs:
        return np.full_like(x, np.nan)
    params = batman.TransitParams()
    params.t0 = t0
    params.per = period
    params.rp = p
    params.a = a_rs
    params.inc = np.degrees(np.arccos(np.clip(b / a_rs, 1e-9, 1.0
        - 1e-9)))
    params.ecc = 0.0
    params.w = 90.0
    params.u = [u1, u2]
    params.limb_dark = "quadratic"
    m = batman.TransitModel(params, x, nthreads=1)
    flux_bat = m.light_curve(params)
    flux_dil = D * flux_bat + (1.0 - D)
    return (flux_dil - 1.0) * 1e3 # ppt
def transit_model(b, p, t0, rho_star, x, u1, u2):
    """Un-diluted alias."""
    return transit_model_diluted(b, p, t0, rho_star, 1.0, x, u1, u2)
#
-----

# Log-posterior -- D_TESS is FIXED to 1.0
# ndim = 3 + 5*N
#
-----

_LOG_SW_LO = np.log(1e-4) # ppt
_LOG_SW_HI = np.log(50.0) # ppt (tightened from 100)
# Fixed TESS dilution
_D_TESS_FIXED = 1.0
# Gaussian prior on impact parameter
_B_PRIOR_MU = 0.3

```

```

_B_PRIOR_SIGMA = 0.10 # tightened from 0.15
def log_probability(theta):
    try:
        theta = np.asarray(theta, dtype=float)
        if not np.isfinite(theta).all():
            return -np.inf
        N = len(DATASETS)
        if len(theta) != 3 + 5 * N:
            return -np.inf
        t0 = theta[0]
        log_rho = theta[1]
        r1_global = theta[2]
        r2_arr = theta[3 : 3 + N]
        q1_arr = theta[3 + N : 3 + 2 * N]
        q2_arr = theta[3 + 2 * N : 3 + 3 * N]
        mean_arr = theta[3 + 3 * N : 3 + 4 * N]
        lsw_arr = theta[3 + 4 * N : 3 + 5 * N]
        rho_star = np.exp(log_rho)
        # Hard domain checks
        if not np.isfinite(rho_star) or rho_star <= 0:
            return -np.inf
        if not (0.0 <= r1_global <= 1.0):
            return -np.inf
        if np.any(r2_arr < 0.0) or np.any(r2_arr > 1.0):
            return -np.inf
        if np.any(q1_arr < 0.0) or np.any(q1_arr > 1.0):
            return -np.inf
        if np.any(q2_arr < 0.0) or np.any(q2_arr > 1.0):
            return -np.inf
        if np.any(lsw_arr < _LOG_SW_LO) or np.any(lsw_arr > _LOG_SW_HI):
            return -np.inf
        lp = 0.0
        # t0: Gaussian (sigma = 2*DUR/24, tighter from 3*DUR/24)
        lp += stats.norm.logpdf(t0, TO_REF, 2.0 * DUR_GUESS / 24.0)
        # log_rho: Gaussian with sigma=0.08 (tightened from 0.12)
        lp += stats.norm.logpdf(log_rho, np.log(RHO_STAR_GUESS), 0.08)
        if not np.isfinite(lp):
            return -np.inf
        # Derive b from r1_global
        b_global, _ = _r1r2_to_bp(r1_global, 0.5)
        if not (0.0 <= b_global < 1.0):
            return -np.inf
        # Gaussian prior on b: N(0.3, 0.10)
        lp += stats.norm.logpdf(b_global, _B_PRIOR_MU, _B_PRIOR_SIGMA)
        if not np.isfinite(lp):
            return -np.inf
        # Dataset likelihoods
        loglike = 0.0
        dataset_list = list(DATASETS.items())
        for i, (name, (x, y, yerr)) in enumerate(dataset_list):
            _, pi = _r1r2_to_bp(r1_global, r2_arr[i])
            if pi <= 0 or pi >= 1.0:
                return -np.inf
            u1_i, u2_i = _q1q2_to_u1u2(q1_arr[i], q2_arr[i])
            mean_i = mean_arr[i]
            sigma_w = np.exp(lsw_arr[i])
            # Dilution: ALL fixed to 1.0
            Di = _D_TESS_FIXED if "TESS" in name.upper() else 1.0

```

```

sigma_mean = max(10.0 * float(np.median(np.abs(yerr))), 1e
-6)
lp += stats.norm.logpdf(mean_i, 0.0, sigma_mean)
if not np.isfinite(lp):
    return -np.inf
model_i = transit_model_diluted(b_global, pi, t0, rho_star,
                                Di, x, u1_i, u2_i)
if not np.all(np.isfinite(model_i)):
    return -np.inf
var_total = yerr**2 + sigma_w**2
resid      = y - (mean_i + model_i)
loglike    += -0.5 * np.sum(resid**2 / var_total +
                            np.log(2.0 * np.pi * var_total))

total = lp + loglike
return float(total) if np.isfinite(total) else -np.inf
except Exception:
    return -np.inf

```

### Listing 12: Datenladen und Vorverarbeitung

```

# Cell 3: Data loading and preprocessing
import numpy as np
import matplotlib.pyplot as plt
from collections import OrderedDict
from pathlib import Path
outdir = Path("output")
outdir.mkdir(exist_ok=True)
def rebase_zero_ppt(time, flux, err, half_width_days=1/24, do_linear=
False):
    time = np.asarray(time, dtype=float)
    flux = np.asarray(flux, dtype=float)
    err = np.asarray(err, dtype=float)
    tmid = time[np.argmin(flux)]
    oot = np.abs(time - tmid) >= half_width_days
    if do_linear and np.sum(oot) >= 2:
        coeffs = np.polyfit(time[oot], flux[oot], 1)
        baseline = np.polyval(coeffs, time)
        flux = flux / baseline
        err = err / np.abs(baseline)
    baseline_med = np.nanmedian(flux[oot]) if np.any(oot) else np.
nanmedian(flux)
    flux0 = (flux / baseline_med - 1.0) * 1e3
    err0 = err / baseline_med * 1e3
    return flux0, err0, baseline_med
# SaintEX
SX_data = np.genfromtxt("target.txt", delimiter=",",
                        names=["BJD", "relflux", "relflux_err"])
time_SX = SX_data["BJD"]
flux_SX_raw = SX_data["relflux"]
flux_err_SX_raw = SX_data["relflux_err"]
flux_SX, flux_err_SX, sx_base = rebase_zero_ppt(
    time_SX, flux_SX_raw, flux_err_SX_raw, do_linear=False)
print(f"SaintEX: {len(time_SX)} pts, baseline={sx_base:.6f}")
# NOVO (differential photometry ~0.011 -> normalize)
NOVO_data = np.genfromtxt("novosibirsk_data.txt", delimiter=",",
                          names=["BJD", "flux", "flux_err"])
time_NOVO = NOVO_data["BJD"]
flux_NOVO_raw = NOVO_data["flux"]
flux_err_NOVO_raw = NOVO_data["flux_err"]

```

```

novo_baseline      = np.median(flux_NOVO_raw)
flux_NOVO_raw      = flux_NOVO_raw      / novo_baseline
flux_err_NOVO_raw  = flux_err_NOVO_raw / novo_baseline
flux_NOVO, flux_err_NOVO, novo_base = rebase_zero_ppt(
    time_NOVO, flux_NOVO_raw, flux_err_NOVO_raw, do_linear=True)
print(f"NOVO: {len(time_NOVO)} pts, raw baseline = {novo_baseline:.6f}")
# TESS (BTJD -> BJD)
TESS_data = np.genfromtxt("TESS_NuanceGP_new_LC.txt", delimiter=None,
    names=["time", "flux", "flux_err"])
time_TESS      = TESS_data["time"] + 2457000.0
flux_TESS_raw  = TESS_data["flux"]
flux_err_TESS_raw = TESS_data["flux_err"]
flux_TESS, flux_err_TESS, tess_base = rebase_zero_ppt(
    time_TESS, flux_TESS_raw, flux_err_TESS_raw, do_linear=False)
print(f"TESS: {len(time_TESS)} pts, BTJD->BJD applied")
# Diagnostic plot
datasets_plot = OrderedDict([
    ("Saint-Ex", (time_SX, flux_SX, flux_err_SX)),
    ("Novosibirsk (LHO)", (time_NOVO, flux_NOVO, flux_err_NOVO)),
    ("TESS", (time_TESS, flux_TESS, flux_err_TESS)),
])
fig, axes = plt.subplots(3, 1, figsize=(13, 10), constrained_layout=True)
for ax, (name, (t, f, fe)) in zip(axes, datasets_plot.items()):
    ax.errorbar(t, f, yerr=fe, fmt=".k", alpha=0.5, ms=2.5, elinewidth=0.5)
    ax.axhline(0, color="gray", lw=1, ls="--")
    ax.set_title(name, fontsize=12)
    ax.set_xlabel("BJD"); ax.set_ylabel("Flux [ppt]"); ax.grid(alpha=0.2)
fig.savefig("output/data_diagnostic.png", dpi=150, bbox_inches="tight")
plt.show()

```

### Listing 13: Initialisierung des Transit-Workers

```

# Cell 4: Initialize transit worker
import numpy as np
from collections import OrderedDict
from transitworker import init_worker, log_probability
DATASETS = OrderedDict([
    ("SaintEX", (time_SX, flux_SX, flux_err_SX)),
    ("NOVO", (time_NOVO, flux_NOVO, flux_err_NOVO)),
    ("TESS", (time_TESS, flux_TESS, flux_err_TESS)),
])
PERIOD_GUESS      = 4.1723
TO_REF            = 2460576.091201
DUR_GUESS         = 1.684
DEPTH_GUESS       = 0.055
RHO_STAR_GUESS    = 5.891
init_worker(DATASETS, PERIOD_GUESS, TO_REF, DUR_GUESS, DEPTH_GUESS,
    RHO_STAR_GUESS)
print(f"init_worker OK | ndim = {3+5*len(DATASETS)} | D_TESS = 1.0")

```

### Listing 14: MCMC-Konfiguration und Ausführung

```

# Cell 5: MCMC setup and run

```

```

# v9: log_rho sigma=0.08, _PU=0.28, b prior N(0.3,0.10), t0 sigma=2*DUR
/24
# nsteps=20000, burn-in=65%, log_sw_hi=ln(50)
import numpy as np
import emcee
from multiprocessing import Pool
from transitworker import log_probability, init_worker, _r1r2_to_bp, _PL
, _PU
N = len(DATASETS)
ndim = 3 + 5 * N # = 18 for N=3 (no D_TESS)
nwalkers = max(4 * ndim, 100)
print(f"ndim={ndim}, nwalkers={nwalkers}")
p_guess = np.sqrt(DEPTH_GUESS) # ~0.2345
print(f"p_guess (Rp/Rs)={p_guess:.4f}")
# ---- Starting values
-----
theta_c = np.zeros(ndim)
theta_c[0] = TO_REF
theta_c[1] = np.log(RHO_STAR_GUESS)
# r1 for b ~ 0.3 (matches b prior center)
Ar = (_PU - _PL) / (2.0 + _PL + _PU)
b_target = 0.3
theta_c[2] = 1.0 - (1.0 - Ar) * (1.0 - b_target / (1.0 + _PL))
# r2 for p ~ p_guess
theta_c[3 : 3 + N] = (p_guess - _PL) / (_PU - _PL)
# Verify
b0, _ = _r1r2_to_bp(theta_c[2], 0.5)
print(f"Initial b={b0:.4f}")
for i, nm in enumerate(DATASETS.keys()):
    _, p0i = _r1r2_to_bp(theta_c[2], theta_c[3+i])
    print(f"Initial Rp/Rs({nm})={p0i:.4f}")
# q1, q2 from PHOENIX LDC for M-dwarf Teff ~3400K
_LDC_INIT = {"Saint-Ex": (0.63, 0.13), "Novisibirsk(LHO)": (0.63, 0.13)
, "TESS": (0.51, 0.19)}
dataset_names = list(DATASETS.keys())
for i, nm in enumerate(dataset_names):
    key = next((k for k in _LDC_INIT if k in nm.upper()), None)
    u1, u2 = _LDC_INIT.get(key, (0.5, 0.1))
    s = u1 + u2
    theta_c[3 + N + i] = s ** 2
    theta_c[3 + 2 * N + i] = 0.5 * u1 / s
# Mean offsets = 0
theta_c[3 + 3 * N : 3 + 4 * N] = 0.0
# log_sw: start at median yerr
for i, (name, (x, y, yerr)) in enumerate(DATASETS.items()):
    sw0 = max(float(np.median(np.abs(yerr))), 1e-4)
    theta_c[3 + 4 * N + i] = np.log(sw0)
# ---- Perturbation widths (tighter for v9)
-----
widths = np.zeros(ndim)
widths[0] = 3e-4 # t0 (tighter)
widths[1] = 0.02 # log_rho (tighter)
widths[2] = 0.02 # r1_global (tighter)
widths[3 : 3 + N] = 0.03 # r2_i (tighter)
widths[3 + N : 3 + 2 * N] = 0.03 # q1_i
widths[3 + 2 * N : 3 + 3 * N] = 0.03 # q2_i
for i, (name, (x, y, yerr)) in enumerate(DATASETS.items()):
    widths[3 + 3 * N + i] = 0.01 * max(float(np.median(np.abs(yerr))), 1

```

```

        e-4)
        widths[3 + 4 * N + i] = 0.06
rng = np.random.default_rng(42)
p0 = theta_c + widths * rng.normal(size=(nwalkers, ndim))
# Clip to prior support
p0[:, 1] = np.clip(p0[:, 1], np.log(0.5), np.log(30.0))
p0[:, 2] = np.clip(p0[:, 2], 0.01, 0.99)
for i in range(N):
    p0[:, 3 + i] = np.clip(p0[:, 3 + i], 0.01, 0.99)
    p0[:, 3 + N + i] = np.clip(p0[:, 3 + N + i], 0.01, 0.99)
    p0[:, 3 + 2 * N + i] = np.clip(p0[:, 3 + 2 * N + i], 0.01, 0.99)
    p0[:, 3 + 4 * N + i] = np.clip(p0[:, 3 + 4 * N + i],
                                    np.log(1e-4), np.log(50.0))
n_valid = sum(1 for p in p0 if np.isfinite(log_probability(p)))
print(f"\nValid starting positions: {n_valid} / {nwalkers}")
# --- MCMC run: 20 000 steps, burn-in = 65%
-----
nsteps = 20000
def _worker_init():
    init_worker(DATASETS, PERIOD_GUESS, TO_REF, DUR_GUESS, DEPTH_GUESS,
                RHO_STAR_GUESS)
with Pool(initializer=_worker_init) as pool:
    sampler = emcee.EnsembleSampler(nwalkers, ndim, log_probability,
                                    pool=pool)
    sampler.run_mcmc(p0, nsteps, progress=True)
chain = sampler.get_chain()
print(f"\nChain shape: {chain.shape}")
af = np.mean(sampler.acceptance_fraction)
print(f"Mean acceptance fraction: {af:.3f} (target: 0.20-0.50)")
nburn = int(nsteps * 0.65)
print(f"Burn-in: first {nburn} steps (65%)")

```

**Listing 15: Trace-Plots zur Konvergenzdiagnose**

```

# Cell 6: Trace plots for convergence diagnostics -- physical parameters
only
import numpy as np
import matplotlib.pyplot as plt
chain = sampler.get_chain() # (nsteps, nwalkers, ndim)
N = len(DATASETS)
nburn = int(chain.shape[0] * 0.65)
dataset_names = list(DATASETS.keys())
# Layout: [t0, ln_rho, r1 | r2xN | q1xN | q2xN | Delta_FxN |
ln_sigma_wxN]
param_indices = (
    [0, 1, 2] # t0, ln rho_*, r1
    + list(range(3, 3 + N)) # r2 per dataset
    # q1, q2, Delta_F, ln sigma_w omitted
)
param_labels = (
    [r"$t_0$ [BJD]", r"$\ln \rho_\star$", r"$r_1$"]
    + [rf"$r_2$\_{\{nm\}}$" for nm in dataset_names]
)
group_starts = {
    0: "Orbital / Stellar",
    3: r"$r_2\_{\{nm\}} R_p / R_\star$ proxy",
}
ncols = 2
nparams = len(param_indices)

```

```

nrows = (nparams + ncols - 1) // ncols
fig, axes = plt.subplots(
    nrows, ncols,
    figsize=(16, 2.0 * nrows),
    sharex=True,
    constrained_layout=True,
)
axes_flat = axes.flatten()
for idx, (pi, lbl) in enumerate(zip(param_indices, param_labels)):
    ax = axes_flat[idx]
    ax.plot(chain[:, ::5, pi], alpha=0.12, lw=0.35)
    ax.axvline(nburn, color="red", ls="--", lw=1.1,
              label=f"burn-in_{nburn}" if idx == 0 else None)
    ax.set_ylabel(lbl, fontsize=8)
    ax.grid(alpha=0.15)
    if pi in group_starts:
        ax.set_title(group_starts[pi], fontsize=8, loc="left",
                    color="steelblue", style="italic", pad=2)
    if idx == 0:
        ax.legend(fontsize=8, loc="upper_right")
for ax in axes_flat[nparams:]:
    ax.set_visible(False)
for ax in axes_flat[nrows * ncols - ncols : nrows * ncols]:
    ax.set_xlabel("Step_number", fontsize=9)
fig.suptitle(
    f"MCMC_Trace_Plots_{nparams}_physikalische_Parameter_(jeder_
    fuenfte_Walker)",
    fontsize=13, y=1.01,
)
fig.savefig("output/trace_plots_v9.png", dpi=150, bbox_inches="tight")
plt.show()
print(f"Post-burn-in_samples: {(chain.shape[0] - nburn) * chain.shape
[1]:,}")

```

**Listing 16:** *Corner-Plot der physikalischen Parameter*

```

# Cell 7: Eckplot -- physikalische Parameter
import numpy as np
import corner
import matplotlib.pyplot as plt
from transitworker import _a_over_rs, _r1r2_to_bp, _q1q2_to_u1u2
chain = sampler.get_chain()
nsteps_run = chain.shape[0]
N = len(DATASETS)
nburn = int(nsteps_run * 0.65)
samples = chain[nburn:, :, :].reshape(-1, ndim)
print(f"Verwendete Stichproben nach Burn-in: {samples.shape[0]:,}")
thin_plot = max(1, samples.shape[0] // 60000)
sp = samples[:, :thin_plot]
print(f"Ausgeduennt auf {sp.shape[0]:,} fuer den Eckplot")
# Dataset-Namen umbenennen: NOVO -> LHO
_raw_names = list(DATASETS.keys())
dataset_names = ["LHO" if "NOVO" in nm.upper() else nm for nm in
_raw_names]
# Physikalische Parameter ableiten
t0_s = sp[:, 0]
rho_s = np.exp(sp[:, 1])
b_s = np.array([_r1r2_to_bp(r1, 0.5)[0] for r1 in sp[:, 2]])
p_s = np.array([_r1r2_to_bp(sp[j, 2], sp[j, 3 + i])[1]

```

```

        for j in range(len(sp))]
        for i in range(N))]
a_rs_s = np.array([_a_over_rs(r, PERIOD_GUESS) for r in rho_s])
inc_s = np.degrees(np.arccos(np.clip(b_s / a_rs_s, 1e-9, 1.0 - 1e-9)))
cols = [t0_s, rho_s, a_rs_s, inc_s, b_s]
lbls = [
    r"$t_0$ [BJD]",
    r"$\rho_\star$ [g cm$^{-3}$]",
    r"$a/R_\star$",
    r"$i$ [deg]",
    r"$b$",
]
]
for i, nm in enumerate(dataset_names):
    cols.append(p_s[i])
    lbls.append(r"$R_p/R_\star$" + f"\n({nm})")
plot_array = np.vstack(cols).T
ranges = []
for j in range(plot_array.shape[1]):
    col = plot_array[:, j]
    lo, hi = np.percentile(col, [0.5, 99.5])
    pad = 0.05 * max(hi - lo, 1e-10)
    ranges.append((lo - pad, hi + pad))
fig = corner.corner(
    plot_array, labels=lbls, range=ranges, bins=40, smooth=1.5,
    show_titles=True, title_fmt=".4f",
    label_kwargs={"fontsize": 7}, title_kwargs={"fontsize": 7},
    quantiles=[0.16, 0.50, 0.84],
)
fig.savefig("output/corner_posterior_v9.png", dpi=200, bbox_inches="
    tight")
plt.show()

```

### Listing 17: Transitmodellplot mit Chi-Quadrat und Residuen

```

# Cell 8: Transitmodell + chi2-Statistik je Datensatz + Residuen
import numpy as np
import matplotlib.pyplot as plt
from transitworker import transit_model_diluted, _r1r2_to_bp,
    _q1q2_to_u1u2
chain = sampler.get_chain()
N = len(DATASETS)
nburn = int(chain.shape[0] * 0.65)
samples = chain[nburn:, :, :].reshape(-1, ndim)
theta_best = np.median(samples, axis=0)
t0_b = theta_best[0]
rho_b = np.exp(theta_best[1])
r1_b = theta_best[2]
b_b, _ = _r1r2_to_bp(r1_b, 0.5)
# Anzeigenamen: NOVO -> Novosibirsk
_raw_names = list(DATASETS.keys())
display_names = ["Novosibirsk" if "NOVO" in nm.upper() else nm for nm in
    _raw_names]
# 2 Zeilen: oben = Daten + Modell, unten = Residuen (halbe Hoehe)
fig, axes = plt.subplots(
    2, N,
    figsize=(5.5 * N, 6.5),
    sharex="col",
    constrained_layout=True,
    gridspec_kw={"height_ratios": [2, 1]},

```

```

)
if N == 1:
    axes = np.array([[axes[0]], [axes[1]]])
dataset_list = list(DATASETS.items())
global_chisq = 0.0
global_n = 0
chi2_per_ds = {}
for i, ((name, (x, y_ppt, yerr_ppt)), disp_name) in enumerate(
    zip(dataset_list, display_names)):
    ax_flux = axes[0, i]
    ax_res = axes[1, i]
    x = np.asarray(x, dtype=float)
    y_ppt = np.asarray(y_ppt, dtype=float)
    yerr_ppt = np.asarray(yerr_ppt, dtype=float)
    _, pi_b = _r1r2_to_bp(r1_b, theta_best[3 + i])
    u1_b, u2_b = _q1q2_to_u1u2(theta_best[3 + N + i], theta_best[3 + 2*N
        + i])
    mean_b = theta_best[3 + 3*N + i]
    sigma_w = np.exp(theta_best[3 + 4*N + i])
    Di_b = 1.0
    mod = transit_model_diluted(b_b, pi_b, t0_b, rho_b, Di_b, x, u1_b,
        u2_b)
    xrel = (x - x.min()) * 24.0
    y0 = y_ppt - mean_b
    if np.all(np.isnan(mod)):
        chi2_red = np.nan; ndof_i = None
        resid = np.full_like(y0, np.nan)
    else:
        resid = y0 - mod
        var_tot = yerr_ppt**2 + sigma_w**2
        chisq_i = np.sum(resid**2 / var_tot)
        k_i = 3 + 5
        ndof_i = max(len(x) - k_i, 1)
        chi2_red = chisq_i / ndof_i
        global_chisq += chisq_i
        global_n += len(x)
        chi2_per_ds[disp_name] = {
            "chi2": chisq_i, "ndof": ndof_i, "chi2_red": chi2_red
        }
    print(f"{disp_name}:  $\chi^2$ ={chisq_i:.1f},  $\nu$ "
        f"Freiheitsgrade={ndof_i},  $\chi^2_{red}$ ={chi2_red:.3f}")
    # Oben: Daten + Modell
    ax_flux.errorbar(
        xrel, y0, yerr=yerr_ppt, fmt=".", color="0.25", ecolor="0.55",
        elinewidth=0.5, capsize=0, ms=3, alpha=0.55, label=disp_name,
    )
    if not np.all(np.isnan(mod)):
        xsort = np.argsort(x)
        ax_flux.plot(xrel[xsort], mod[xsort],
            color="red", lw=2, label="Bestes Modell")
    titel = (f"{disp_name}\n $\chi^2$ \n $\nu$ ={chi2_red:.2f}")
        if ndof_i else f"{disp_name}\n(kein Modell)")
    ax_flux.set_title(titel, fontsize=11)
    ax_flux.grid(alpha=0.18)
    if i == 0:
        ax_flux.set_ylabel("Relative Helligkeit [ppt]")
    ax_flux.legend(fontsize=9)
    # Unten: Residuen

```

```

if not np.all(np.isnan(mod)):
    ax_res.axhline(0.0, color="red", lw=1.5, alpha=0.9, label="
        Modell")
else:
    ax_res.axhline(0.0, color="k", lw=1, alpha=0.5)
ax_res.errorbar(
    xrel, resid, yerr=yerr_ppt, fmt=".", color="0.25",
    ecolor="0.55", elinewidth=0.5, capsize=0, ms=3, alpha=0.7,
    label="Residuen",
)
ax_res.set_xlabel("Zeit seit Beobachtungsbeginn [h]")
if i == 0:
    ax_res.set_ylabel("Residuen [ppt]")
ax_res.grid(alpha=0.18)
if i == 0:
    ax_res.legend(fontsize=8)
global_ndof = max(global_n - ndim, 1)
global_chi2r = global_chisq / global_ndof
chi2_per_ds["Global"] = {
    "chi2": global_chisq, "ndof": global_ndof, "chi2_red": global_chi2r
}
print(f"\nGesamt: chi2={global_chisq:.1f},
      f"Freiheitsgrade={global_ndof}, chi2_red={global_chi2r:.3f}")
fig.savefig("output/transitmodell_residuen_v10.png", dpi=200,
            bbox_inches="tight")
plt.show()

```

### Listing 18: Phasengefaltete Lichtkurven mit gebinnten Punkten

```

# Cell 10: Phasengefalteter Transit + Residuen je Datensatz (mit
# gebinnten Punkten)
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.gridspec import GridSpec
from transitworker import transit_model_diluted, _r1r2_to_bp,
    _q1q2_to_u1u2
# -- Posteriors
-----
chain = sampler.get_chain()
N = len(DATASETS)
nburn = int(chain.shape[0] * 0.65)
samples = chain[nburn:, :, :].reshape(-1, ndim)
theta_best = np.median(samples, axis=0)
t0_b = theta_best[0]
rho_b = np.exp(theta_best[1])
r1_b = theta_best[2]
b_b, _ = _r1r2_to_bp(r1_b, 0.5)
PERIOD = 4.1723
N_MODEL = 1000
NBINS = 500
# Anzeigenamen: NOVO -> Novosibirsk
_raw_names = list(DATASETS.keys())
display_names = ["Novosibirsk" if "NOVO" in nm.upper() else nm for nm in
    _raw_names]
# -- Phasenfaltungs-Hilfsfunktion
-----
def phase_fold(t, t0, period):
    """Gibt Phase in [-0.5, 0.5] zurueck."""
    phi = ((t - t0) / period) % 1.0

```

```

    phi[phi > 0.5] -= 1.0
    return phi
# -- Abbildungs-Layout
-----
fig = plt.figure(figsize=(5.8 * N, 7.0), facecolor="white")
fig.subplots_adjust(hspace=0.0)
dataset_list = list(DATASETS.items())
colors_ds    = ["#3a86ff", "#ff6b6b", "#2ec4b6"]
for i, ((name, (x, y_ppt, yerr_ppt)), disp_name) in enumerate(
    zip(dataset_list, display_names)):
    gs_col = GridSpec(
        2, N, figure=fig,
        left=0.07, right=0.97,
        top=0.93, bottom=0.09,
        hspace=0.0, wspace=0.28,
        height_ratios=[3, 1],
    )
    ax_flux = fig.add_subplot(gs_col[0, i])
    ax_res  = fig.add_subplot(gs_col[1, i], sharex=ax_flux)
    x        = np.asarray(x, dtype=float)
    y_ppt    = np.asarray(y_ppt, dtype=float)
    yerr_ppt = np.asarray(yerr_ppt, dtype=float)
    _, pi_b  = _r1r2_to_bp(r1_b, theta_best[3 + i])
    u1_b, u2_b = _q1q2_to_u1u2(theta_best[3 + N + i], theta_best[3 + 2*N
        + i])
    mean_b    = theta_best[3 + 3*N + i]
    sigma_w   = np.exp(theta_best[3 + 4*N + i])
    Di_b     = 1.0
# -- Phasenfaltung
-----
phi_data = phase_fold(x, t0_b, PERIOD)
phi_h    = phi_data * PERIOD * 24.0
y0       = y_ppt - mean_b
# -- Dichtes Modellgitter
-----
phi_model = np.linspace(-0.5, 0.5, N_MODEL)
t_model   = t0_b + phi_model * PERIOD
mod_dense = transit_model_diluted(
    b_b, pi_b, t0_b, rho_b, Di_b, t_model, u1_b, u2_b
)
phi_model_h = phi_model * PERIOD * 24.0
# -- Residuen und chi2_nu
-----
mod_at_data = transit_model_diluted(
    b_b, pi_b, t0_b, rho_b, Di_b, x, u1_b, u2_b
)
if not np.all(np.isnan(mod_at_data)):
    resid    = y0 - mod_at_data
    var_tot  = yerr_ppt**2 + sigma_w**2
    chisq_i  = np.sum(resid**2 / var_tot)
    ndof_i   = max(len(x) - 8, 1)
    chi2_red = chisq_i / ndof_i
    print(f"{disp_name}: □ chi2={chisq_i:.1f} □ □ Freiheitsgrade={ndof_i}
        □ □"
          f" chi2_nu={chi2_red:.3f}")
else:
    resid    = np.full_like(y0, np.nan)
    chi2_red = np.nan

```

```

        ndof_i = None
    col = colors_ds[i % len(colors_ds)]
    # -- Gebinnte Datenpunkte
    -----
    bins = np.linspace(-0.5, 0.5, NBINS + 1)
    bin_idx = np.digitize(phi_data, bins) - 1
    phi_bin, y_bin, yerr_bin = [], [], []
    for k in range(NBINS):
        m = bin_idx == k
        if np.any(m):
            phi_bin.append(np.mean(phi_data[m]))
            y_bin.append(np.mean(y0[m]))
            yerr_bin.append(np.sqrt(np.mean(yerr_ppt[m]**2) / np.sum(m)))
    phi_bin = np.array(phi_bin)
    y_bin = np.array(y_bin)
    yerr_bin = np.array(yerr_bin)
    phi_bin_h = phi_bin * PERIOD * 24.0
    # -- Oberes Panel: Daten + gebinnte Punkte + Modell
    -----
    ax_flux.errorbar(
        phi_h, y0, yerr=yerr_ppt,
        fmt="o", ms=3.0, color=col, alpha=0.25,
        ecolor=col, elinewidth=0.5, capsize=0,
        label=disp_name, zorder=1,
    )
    if phi_bin.size > 0:
        ax_flux.errorbar(
            phi_bin_h, y_bin, yerr=yerr_bin,
            fmt="o", ms=5.0, color="#ffb000", ecolor="#ffb000",
            elinewidth=1.2, capsize=0, alpha=0.95,
            label="Gebinnt", zorder=3,
        )
    if not np.all(np.isnan(mod_dense)):
        ax_flux.plot(
            phi_model_h, mod_dense,
            color="crimson", lw=2.0, zorder=4, label="Bestes Modell",
        )
    ax_flux.axvline(0.0, color="k", lw=0.8, ls=":", alpha=0.4, zorder=0)
    ax_flux.set_xlim(-2, 2)
    ax_flux.set_ylim(-100, 100)
    chi2_label = (f"$\\chi^2_{\\nu} \\{chi2_red:.2f}$"
                 if ndof_i else "(kein Modell)")
    ax_flux.set_title(f"{disp_name} -- {chi2_label}", fontsize=11, pad=6)
    ax_flux.set_ylabel("Relative Helligkeit [ppt]", fontsize=10)
    ax_flux.grid(True, alpha=0.15, lw=0.5)
    ax_flux.tick_params(labelbottom=False)
    ax_flux.legend(fontsize=8.0, framealpha=0.7, loc="upper right")
    # -- Unteres Panel: Residuen
    -----
    ax_res.axhline(0.0, color="crimson", lw=1.6, alpha=0.85, zorder=3,
                  label="Nulllinie")
    ax_res.errorbar(
        phi_h, resid, yerr=yerr_ppt,
        fmt="o", ms=3.0, color=col, alpha=0.55,
        ecolor=col, elinewidth=0.6, capsize=0, zorder=2,
    )

```

```

ax_res.axvline(0.0, color="k", lw=0.8, ls=":", alpha=0.4, zorder=1)
ax_res.set_xlim(-2, 2)
ax_res.set_ylim(-100, 100)
ax_res.set_xlabel("Orbitalphase [h]", fontsize=10)
ax_res.set_ylabel("Resid. [ppt]", fontsize=9)
ax_res.grid(True, alpha=0.15, lw=0.5)
ax_res.yaxis.set_major_locator(plt.MaxNLocator(3, symmetric=True))
plt.setp(ax_flux.get_xticklabels(), visible=False)
fig.savefig("output/phasengefalteter_transit_residuen.png", dpi=200,
            bbox_inches="tight")
plt.show()
print("Gespeichert -> output/phasengefalteter_transit_residuen.png")

```

**Listing 19:** *Rp/Rs-Posterior-KDE je Datensatz*

```

# Cell 9: Rp/Rs-Posterior-KDE je Datensatz
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import gaussian_kde
from transitworker import _r1r2_to_bp
chain = sampler.get_chain()
N = len(DATASETS)
nburn = int(chain.shape[0] * 0.65)
samples = chain[nburn:, :, :].reshape(-1, ndim)
_raw_names = list(DATASETS.keys())
display_names = ["Novosibirsk" if "NOV0" in nm.upper() else nm for nm in
                 _raw_names]
colors = ["#1f77b4", "#ff7f0e", "#2ca02c"]
r1_s = samples[:, 2]
fig, ax = plt.subplots(figsize=(6, 4), constrained_layout=True)
xmin_global = np.inf
xmax_global = -np.inf
curves = []
for i, disp_name in enumerate(display_names):
    p_arr = np.array([_r1r2_to_bp(r1_s[j], samples[j, 3 + i])[1]
                     for j in range(len(samples))])
    kde = gaussian_kde(p_arr, bw_method="scott")
    lo = np.percentile(p_arr, 0.1)
    hi = np.percentile(p_arr, 99.9)
    xs = np.linspace(lo, hi, 500)
    xmin_global = min(xmin_global, lo)
    xmax_global = max(xmax_global, hi)
    curves.append((xs, kde, p_arr, disp_name, colors[i]))
span = xmax_global - xmin_global
ax.set_xlim(xmin_global - 0.03 * span, xmax_global + 0.03 * span)
for xs, kde, p_arr, disp_name, col in curves:
    ax.fill_between(xs, kde(xs), alpha=0.20, color=col)
    ax.plot(xs, kde(xs), lw=2, color=col, label=disp_name)
    ax.axvline(np.median(p_arr), ls="--", lw=1.2, color=col)
ax.set_xlabel(r"$R_p/R_\star$", fontsize=13)
ax.set_ylabel("Posteriordichte", fontsize=12)
ax.set_title(r"$R_p/R_\star$-Posterior je Datensatz", fontsize=12)
ax.grid(alpha=0.2)
ax.legend(frameon=True, fontsize=10)
fig.savefig("output/rprs_posteriors_v9.png", dpi=200)
plt.show()

```

**Listing 20:** *Parametertabellen in Publikationsqualitaet*

```

# Cell 11: Publication-quality parameter tables
import numpy as np
import pandas as pd
from IPython.display import display, HTML
from transitworker import _a_over_rs, _r1r2_to_bp, _q1q2_to_u1u2
# ---- Extract posteriors
-----
chain      = sampler.get_chain()
N          = len(DATASETS)
nburn      = int(chain.shape[0] * 0.65)
samples    = chain[nburn:, :, :].reshape(-1, ndim)
dataset_names = list(DATASETS.keys())
def q163(x):
    return np.percentile(x, [16, 50, 84])
def fmt_pm(arr, prec=5):
    """Format median +/- 1-sigma asymmetric."""
    q16, q50, q84 = q163(arr)
    em = q50 - q16
    ep = q84 - q50
    return f"${q50:.{prec}f}^{+{ep:.{prec}f}}_{-{em:.{prec}f}}$"
# Stellar constants
R_star_Rsun = 0.488267
M_star_Msun = 0.4863
R_star_Rearth = R_star_Rsun * 109.1
R_star_m      = R_star_Rsun * 6.957e8
M_star_kg     = M_star_Msun * 1.989e30
R_sun_m       = 6.957e8
R_jup_m       = 7.1492e7
# Physical posterior arrays
rho_s = np.exp(samples[:, 1])
r1_s  = samples[:, 2]
b_s   = np.array([_r1r2_to_bp(r, 0.5)[0] for r in r1_s])
a_rs_s = np.array([_a_over_rs(r, PERIOD_GUESS) for r in rho_s])
inc_s  = np.degrees(np.arccos(np.clip(b_s / a_rs_s, 1e-9, 1.0 - 1e-9)))
a_AU_s = a_rs_s * R_star_m / 1.496e11
# Per-dataset Rp/Rs
p_per_ds = []
for i in range(N):
    p_arr = np.array([_r1r2_to_bp(r1_s[j], samples[j, 3 + i])[1]
                      for j in range(len(samples))])
    p_per_ds.append(p_arr)
# Combined Rp/Rs (inverse-variance weighted)
p_vars = [np.var(pa) for pa in p_per_ds]
weights = np.array([1.0/v if v > 0 else 0 for v in p_vars])
w_sum   = weights.sum()
p_comb  = np.zeros(len(samples))
for pa, w in zip(p_per_ds, weights):
    p_comb += w * pa
p_comb /= w_sum
Rp_earth_s = p_comb * R_star_Rearth
Rp_jup_s   = p_comb * R_star_m / R_jup_m
delta_s    = p_comb**2 * 1e6
P_sec     = PERIOD_GUESS * 86400.0
T14_s     = (P_sec / np.pi) * np.arcsin(
    (1.0 / a_rs_s) * np.sqrt((1.0 + p_comb)**2 - b_s**2) /
    np.sin(np.radians(inc_s))
) / 3600.0
T_eff_star = 3392.0

```

```

T_eq_s = T_eff_star * np.sqrt(R_star_Rsun * R_sun_m / (2.0 * a_rs_s *
    R_star_m))
L_star = (R_star_Rsun)**2 * (T_eff_star / 5778.0)**4
S_inc_s = L_star / a_AU_s**2
# TABLE 1: Stellar Parameters
table1_data = [
    ["TIC_ID", "275717567", "TICv8"],
    ["TOI_ID", "7265.01", "ExoFOP"],
    [" $T_{\text{eff}}$  [K]", "3392 $\pm$ 157", "TICv8"],
    [" $\log g$  [cgs]", "4.748 $\pm$ 0.05", "TICv8"],
    [" $R_{\text{star}}$  [ $R_{\text{star}}$ ]", "0.4883 $\pm$ 0.014", "TICv8"],
    [" $M_{\text{star}}$  [ $M_{\text{star}}$ ]", "0.4863 $\pm$ 0.015", "TICv8"],
    [" $\rho_{\text{star}}$  [ $\text{g cm}^{-3}$ ]", "5.891 $\pm$ 0.50", "TICv8/Sozzetti+2007"],
    ["[Fe/H] [dex]", "0.0 $\pm$ (assumed)", "---"],
    ["Spectral type", "M3V", "TICv8"],
]
df1 = pd.DataFrame(table1_data, columns=["Parameter", "Value", "Source"])
display(HTML("<h3>Table 1: Stellar Parameters (Fixed Inputs)</h3>" + df1
    .to_html(index=False, escape=False)))
# TABLE 2: Global Joint-Fit Parameters
table2_data = []
table2_data.append([" $P$  [days]", "f"{PERIOD_GUESS
    :.6f}", "Fixed"])
table2_data.append([" $e$ ", "0 $\pm$ (fixed)",
    "Circular orbit"])
table2_data.append([" $\omega$  [deg]", "90 $\pm$ (fixed)",
    "Fixed"])
table2_data.append([" $D_{\text{TESS}}$ ", "1.0 $\pm$ (fixed)",
    "No dilution"])
table2_data.append([" $T_0$  [BJD $_{\text{TDB}}$ ]", "fmt_pm(samples
    [:, 0], 5), "MCMC"])
table2_data.append([" $\rho_{\text{star}}$  [ $\text{g cm}^{-3}$ ]", "fmt_pm(rho_s, 3)
    , "MCMC"])
table2_data.append([" $a/R_{\text{star}}$ ", "fmt_pm(a_rs_s,
    3), "Derived"])
table2_data.append([" $i$  [deg]", "fmt_pm(inc_s,
    2), "Derived"])
table2_data.append([" $b$ ", "fmt_pm(b_s, 3),
    "MCMC"])
table2_data.append([" $R_p/R_{\text{star}}$  (combined)", "fmt_pm(p_comb,
    5), "Weighted"])
table2_data.append([" $R_p/R_{\text{earth}}$ ", "fmt_pm(
    Rp_earth_s, 2), "Derived"])
table2_data.append([" $R_p/R_J$ ", "fmt_pm(Rp_jup_s
    , 4), "Derived"])
table2_data.append([" $\Delta$  [ppm]", "fmt_pm(delta_s,
    0), "Derived"])
table2_data.append([" $T_{14}$  [hours]", "fmt_pm(T14_s,
    3), "Derived"])
df2 = pd.DataFrame(table2_data, columns=["Parameter", "Value", "Note"])
display(HTML("<h3>Table 2: Global Joint-Fit Transit & Orbital Parameters
    </h3>" + df2.to_html(index=False, escape=False)))
# TABLE 3: Per-Instrument Parameters
table3_rows = []
for i, nm in enumerate(dataset_names):
    Rp_s_i = p_per_ds[i] * R_star_Rearth

```

```

sw_s_i    = np.exp(samples[:, 3 + 4*N + i])
q1_s_i    = samples[:, 3 + N + i]
q2_s_i    = samples[:, 3 + 2*N + i]
mean_s_i  = samples[:, 3 + 3*N + i]
u1_s_i    = 2.0 * np.sqrt(q1_s_i) * q2_s_i
u2_s_i    = np.sqrt(q1_s_i) * (1.0 - 2.0 * q2_s_i)
chi2_info = chi2_per_ds.get(nm, {})
chi2_str  = f"{chi2_info.get('chi2_red', np.nan):.3f}" if chi2_info
           else "---"
n_pts     = len(DATASETS[nm][0])
table3_rows.append({
    "Instrument": nm,
    "$N_{\mathrm{pts}}$": str(n_pts),
    "$R_p/R_{\star}$": fmt_pm(p_per_ds[i], 5),
    "$R_p$_{\oplus}$": fmt_pm(Rp_s_i, 2),
    "$u_1$": fmt_pm(u1_s_i, 3),
    "$u_2$": fmt_pm(u2_s_i, 3),
    "$q_1$": fmt_pm(q1_s_i, 3),
    "$q_2$": fmt_pm(q2_s_i, 3),
    "$\sigma_w$_{ppt}": fmt_pm(sw_s_i, 3),
    "$m$_{ppt}": fmt_pm(mean_s_i, 3),
    "$\chi^2_{\nu}$": chi2_str,
})
df3 = pd.DataFrame(table3_rows)
display(HTML("<h3>Table 3: Per-Instrument Transit Parameters</h3>" + df3
    .to_html(index=False, escape=False)))
chi2_glob = chi2_per_ds.get("Global", {})
if chi2_glob:
    print(f"\nGlobal chi2_red = {chi2_glob['chi2_red']:.3f} and
        f"(chi2 = {chi2_glob['chi2']:.1f}, ndof = {chi2_glob['ndof']})")
# TABLE 4: Derived Physical Parameters
table4_data = [
    ["$a$_{AU}", fmt_pm(a_AU_s, 5), "From $a$
    /$R_{\star}$ and $R_{\star}$"],
    ["$T_{\mathrm{eq}}$_{K} ($A_B=0$)", fmt_pm(T_eq_s, 1), "$T_{\star} \sqrt{R_{\star}/2a}$"],
    ["$S_{\mathrm{inc}}$_{\oplus}$", fmt_pm(S_inc_s, 2), "$L_{\star}/a^2$"],
    ["$\delta$_{ppm}", fmt_pm(delta_s, 0), "$ (R_p/R_{\star})^2 \times 10^6$"],
    ["$T_{14}$_{hours} (2010), circular", fmt_pm(T14_s, 3), "Winn (2010), circular"],
    ["$R_p/R_{\star}$ (combined)", fmt_pm(p_comb, 5), "Inv-variance weighted"],
    ["$R_p$_{\oplus}$", fmt_pm(Rp_earth_s, 2), "Combined"],
    ["$R_p$_{J$}", fmt_pm(Rp_jup_s, 4), "Combined"],
]
df4 = pd.DataFrame(table4_data, columns=["Parameter", "Value", "Note"])
display(HTML("<h3>Table 4: Derived Physical Parameters</h3>" + df4
    .to_html(index=False, escape=False)))
# TABLE 5: Prior Specification Summary
table5_data = [
    ["$T_0$", "$\mathcal{N}(T_0, \mathrm{ref})_{2T_{14}/24}$", "Gaussian"],
    ["$\ln \rho_{\star}$", "$\mathcal{N}(\ln 5.891, 0.08)$", "Gaussian"],
]

```

```

        "Gaussian"],
["$r_{1}$_{(global)}", "$\\mathcal{U}(0,1)$_{+}$\\mathcal{N}(0.3,\\, ,_{0.10)}$_{on}$b$", "Uniform_{+}Gaussian_{on}derived_{b}$"],
["$r_{2,i}$", "$\\mathcal{U}(0,1)$",
"$p_L=0.03$, $p_U=0.28$"],
],
["$q_{1,i}$", "$\\mathcal{U}(0,1)$",
"Kipping_{(2013)}"],
["$q_{2,i}$", "$\\mathcal{U}(0,1)$",
"Kipping_{(2013)}"],
["$m_i$_{ppt}", "$\\mathcal{N}(0,\\, ,_{10}\\cdot\\mathrm{med}\\sigma_i)$", "Per-dataset_{offset}"],
["$\\ln\\sigma_{w,i}$", "$\\mathcal{U}(\\ln 10^{-4},\\, ,_{\\ln 50})$",
"Log-uniform_{jitter}"],
["$P$_{days}", "4.172300_{(fixed)}",
"TESS/ExoFOP"],
["$e$", "0_{(fixed)}",
"Circular"],
["$D_{\\mathrm{TESS}}$", "1.0_{(fixed)}",
"No_{third-light}],
]
df5 = pd.DataFrame(table5_data, columns=["Parameter", "Prior", "Note"])
display(HTML("<h3>Table_5: Prior Specification</h3>" + df5.to_html(index=False, escape=False)))
print("\\nAll_{tables}_{rendered}.")

```

### Listing 21: CSV-Export der Ergebnisse

```

# Cell 12: Export parameters to CSV
import numpy as np
import pandas as pd
from transitworker import _a_over_rs, _r1r2_to_bp
chain = sampler.get_chain()
N = len(DATASETS)
nburn = int(chain.shape[0] * 0.65)
samples = chain[nburn:, :, :].reshape(-1, ndim)
dataset_names = list(DATASETS.keys())
def q163(x):
    return np.percentile(x, [16, 50, 84])
R_star_Rsun = 0.488267
R_star_Rearth = R_star_Rsun * 109.1
R_star_m = R_star_Rsun * 6.957e8
R_jup_m = 7.1492e7
R_sun_m = 6.957e8
rho_s = np.exp(samples[:, 1])
r1_s = samples[:, 2]
b_s = np.array([_r1r2_to_bp(r, 0.5)[0] for r in r1_s])
a_rs_s = np.array([_a_over_rs(r, PERIOD_GUESS) for r in rho_s])
inc_s = np.degrees(np.arccos(np.clip(b_s / a_rs_s, 1e-9, 1.0 - 1e-9)))
a_AU_s = a_rs_s * R_star_m / 1.496e11
p_per_ds = []
for i in range(N):
    p_arr = np.array([_r1r2_to_bp(r1_s[j], samples[j, 3 + i])[1]
                     for j in range(len(samples))])
    p_per_ds.append(p_arr)
p_vars = [np.var(pa) for pa in p_per_ds]
weights = np.array([1.0/v if v > 0 else 0 for v in p_vars])
w_sum = weights.sum()
p_comb = np.zeros(len(samples))

```

```

for pa, w in zip(p_per_ds, weights):
    p_comb += w * pa
p_comb /= w_sum
Rp_earth_s = p_comb * R_star_Rearth
Rp_jup_s = p_comb * R_star_m / R_jup_m
T_eff_star = 3392.0
T_eq_s = T_eff_star * np.sqrt(R_star_Rsun * R_sun_m / (2.0 * a_rs_s *
    R_star_m))
L_star = (R_star_Rsun)**2 * (T_eff_star / 5778.0)**4
S_inc_s = L_star / a_AU_s**2
delta_s = p_comb**2 * 1e6
P_sec = PERIOD_GUESS * 86400.0
T14_s = (P_sec / np.pi) * np.arcsin(
    (1.0 / a_rs_s) * np.sqrt((1.0 + p_comb)**2 - b_s**2) /
    np.sin(np.radians(inc_s))
) / 3600.0
rows = []
for label, arr, prec in [
    ("T0_□[BJD]", samples[:, 0], 5),
    ("rho_star_□[g/cm3]", rho_s, 3),
    ("a/R_star", a_rs_s, 3),
    ("inc_□[deg]", inc_s, 2),
    ("b", b_s, 3),
    ("Rp/Rs_□(combined)", p_comb, 5),
    ("Rp_□[R_earth]", Rp_earth_s, 2),
    ("Rp_□[R_jup]", Rp_jup_s, 4),
    ("delta_□[ppm]", delta_s, 0),
    ("T14_□[hours]", T14_s, 3),
    ("a_□[AU]", a_AU_s, 5),
    ("T_eq_□[K]", T_eq_s, 1),
    ("S_inc_□[S_earth]", S_inc_s, 2),
]:
    q16, q50, q84 = q163(arr)
    rows.append({"Parameter": label, "Median": round(q50, prec),
        "-1sigma": round(q50-q16, prec), "+1sigma": round(q84-
            q50, prec)})
for i, nm in enumerate(dataset_names):
    Rp_i = p_per_ds[i] * R_star_Rearth
    sw_i = np.exp(samples[:, 3 + 4*N + i])
    q1_i = samples[:, 3 + N + i]
    q2_i = samples[:, 3 + 2*N + i]
    u1_i = 2.0 * np.sqrt(q1_i) * q2_i
    u2_i = np.sqrt(q1_i) * (1.0 - 2.0 * q2_i)
    for label, arr, prec in [
        (f"Rp/Rs_□({nm})", p_per_ds[i], 5),
        (f"Rp_□[R_earth]_□({nm})", Rp_i, 2),
        (f"u1_□({nm})", u1_i, 3),
        (f"u2_□({nm})", u2_i, 3),
        (f"sigma_w_□[ppt]_□({nm})", sw_i, 3),
    ]:
        q16, q50, q84 = q163(arr)
        rows.append({"Parameter": label, "Median": round(q50, prec),
            "-1sigma": round(q50-q16, prec), "+1sigma": round(
                q84-q50, prec)})
df_out = pd.DataFrame(rows)
df_out.to_csv("output/T0I7265_parameters_v9.csv", index=False)
print("Saved: □output/T0I7265_parameters_v9.csv")
print(df_out.to_string(index=False))

```

## 13.2 Python-Code: TRICERATOPS-Validierung

Der folgende Python-Code führt die statistische Validierung von TOI-7265.01 mit dem Werkzeug TRICERATOPS (Giacalone et al., 2021) durch. Die Analyse wird sowohl ohne als auch mit einer Contrast Curve aus der LHO-Nowosibirsk-Beobachtung ausgeführt, um den Einfluss bodengebundener Nachbeobachtungen auf die False Positive Probability (FPP) zu quantifizieren.

**Listing 22:** TRICERATOPS-Validierung fuer TOI-7265.01

```
# =====
# TRICERATOPS-Validierung fuer TOI-7265.01 / TIC 275717567
# =====
# Berechnet die False Positive Probability (FPP) und die
# Nearby False Positive Probability (NFPP) mittels bayesianischer
# Analyse aller moeglichen Transit-Szenarien.
#
# Validierungskriterien (Giacalone et al. 2021):
#   FPP < 0.015 (1.5%) UND
#   NFPP < 0.001 (0.1%) --> Validierter Planet
# =====
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import triceratops.triceratops as tr
from lightkurve import TessLightCurve

# --- Transitparameter aus dem Joint Fit ---
TIC_ID      = 275717567          # TESS Input Catalog ID
SEKTOREN    = np.array([76])    # TESS-Sektor 76 (200s QLP-Kadenz)
P_ORB       = 4.1723            # Orbitalperiode [Tage]
TO_BTJD     = 3564.7804         # Transitmitte [BTJD]
TIEFE_FRAC = 0.039             # Transittiefe als Bruchteil (39 ppt)
# --- Pfad zur Contrast Curve (LHO Nowosibirsk, i'-Band) ---
CC_DATEI    = "contrast_curve_lho.csv"

# =====
# Schritt 1: TRICERATOPS-Zielobjekt initialisieren
# =====
# Fragt den TESS Input Catalog (TIC) nach Nachbarsternen ab
# und laedt die TESS-Pixeldaten (TessCut) fuer den Sektor.
# search_radius = 5 Pixel entspricht ca. 105 Bogensekunden.

ziel = tr.target(
    ID=TIC_ID,
    sectors=SEKTOREN,
    search_radius=5
)
print(f"Gefundene Nachbarsterne: {len(ziel.stars)}")

# =====
# Schritt 2: Sternfeld visualisieren
# =====
# Zeigt die Positionen aller TIC-Sterne relativ zu den
# TESS-Pixeln. Die grosse Pixelskala (21"/Pixel) fuehrt dazu,
# dass mehrere Sterne innerhalb der Apertur liegen.
```

```

ziel.plot_field(sector=SEKTOREN[0])
plt.savefig("triceratops_field.png", dpi=150)
plt.close()

# =====
# Schritt 3: Transittiefen pro Stern berechnen
# =====
# Fuer jeden Stern in der Naehel wird berechnet, welche
# Transittiefe er haben muesste, wenn ER die Quelle des
# beobachteten Signals waere. Dies beruecksichtigt die
# Flussbeitraege aller Sterne in der Apertur (PSF-Modell).

ziel.calc_depths(tdepth=TIEFE_FRAC)

# Flussanteile anzeigen: Der Zielstern traegt nur ~4.3%
# des Gesamtflusses in der TESS-Apertur bei.
print(ziel.stars[["ID", "Tmag", "fluxratio", "sep_(arcsec)"].head(10))

# =====
# Schritt 4: Phasen-gefaltete Lichtkurve vorbereiten
# =====
# TRICERATOPS benoetigt eine phasen-gefaltete, auf ~100 Punkte
# gebinnte Lichtkurve als Eingabe.

# Lichtkurve laden (QLP-Daten, alle Sektoren)
lk_daten = pd.read_csv("tess_lichtkurve.csv")
zeit = lk_daten["time"].values # BTJD
fluss = lk_daten["flux"].values # Normierter Fluss
fehler = lk_daten["flux_err"].values # Flussfehler

# Phasenfaltung: Transit auf Phase 0 zentrieren
phase = ((zeit - TO_BTJD) % P_ORB) / P_ORB
phase[phase > 0.5] -= 1.0
zeit_gefaltet = phase * P_ORB # Tage ab Transitmitte

# Nur Daten nahe des Transits auswaehlen (+/- 3.6 Stunden)
fenster = 0.15 # Tage
maske = np.abs(zeit_gefaltet) < fenster
t_transit = zeit_gefaltet[maske]
f_transit = fluss[maske]
e_transit = fehler[maske]

# Chronologisch sortieren
idx = np.argsort(t_transit)
t_transit = t_transit[idx]
f_transit = f_transit[idx]
e_transit = e_transit[idx]

# Auf ~100 Punkte binnen (empfohlen fuer TRICERATOPS)
lk_obj = TessLightCurve(time=t_transit, flux=f_transit, flux_err=
    e_transit)
bin_groesse = (t_transit.max() - t_transit.min()) / 100
lk_gebinnt = lk_obj.bin(time_bin_size=bin_groesse)
t_bin = lk_gebinnt.time.value
f_bin = lk_gebinnt.flux.value
e_bin = np.mean(lk_gebinnt.flux_err.value)
print(f"Gebinnte_Lichtkurve:_{len(t_bin)}_Punkte")

```

```

# =====
# Schritt 5: Szenario-Wahrscheinlichkeiten OHNE Contrast Curve
# =====
# TRICERATOPS bewertet folgende Szenarien:
# TP      = Transitierender Planet auf dem Zielstern
# EB      = Eklipsierender Doppelstern auf dem Zielstern
# EBx2P   = EB mit doppelter Periode
# PTP     = Planet auf physisch gebundenem Begleiter
# PEB     = EB auf physisch gebundenem Begleiter
# STP     = Planet auf Sekundaerstern
# DTP     = Planet auf Doppelstern
# BTP     = Planet auf Hintergrundstern
# BEB     = EB auf Hintergrundstern
# NTP     = Planet auf aufgeluestem Nachbarstern
# NEB     = EB auf aufgeluestem Nachbarstern
# (jeweils auch mit EBx2P-Varianten)

ziel.calc_probs(
    time=t_bin,
    flux_0=f_bin,
    flux_err_0=e_bin,
    P_orb=P_ORB,
    parallel=True
)

fpp_ohne = ziel.FPP
nfpp_ohne = ziel.NFPP
print(f"Ohne Contrast Curve: FPP={fpp_ohne:.4f}, NFPP={nfpp_ohne:.4f}")

# =====
# Schritt 6: Szenario-Wahrscheinlichkeiten MIT Contrast Curve
# =====
# Die Contrast Curve aus der LHO-Nowosibirsk-Beobachtung
# (0.28m-Teleskop, i'-Band, FWHM = 1.9") beschreibt, wie
# lichtschwach ein Begleiter bei gegebenem Winkelabstand
# noch detektierbar ist. Damit koennen unaufgeloeste
# Kontaminationsquellen innerhalb der PSF eingeschaenkt werden.

ziel.calc_probs(
    time=t_bin,
    flux_0=f_bin,
    flux_err_0=e_bin,
    P_orb=P_ORB,
    parallel=True,
    contrast_curve_file=CC_DATEI,
    filt="TESS"
)

fpp_mit = ziel.FPP
nfpp_mit = ziel.NFPP
print(f"Mit Contrast Curve: FPP={fpp_mit:.4f}, NFPP={nfpp_mit:.4f}")

# =====
# Schritt 7: Robuste Schaeztung ueber mehrere Durchlaeufe
# =====
# Giacalone et al. (2021) empfehlen, calc_probs mehrfach

```

```

# auszufuehren und Mittelwert/Standardabweichung anzugeben,
# da die Monte-Carlo-Simulation inhaerent stochastisch ist.
N_ITER = 5
fpp_liste = np.zeros(N_ITER)
nfpp_liste = np.zeros(N_ITER)
for i in range(N_ITER):
    ziel.calc_probs(
        time=t_bin,
        flux_0=f_bin,
        flux_err_0=e_bin,
        P_orb=P_ORB,
        parallel=True,
        verbose=0,
        contrast_curve_file=CC_DATEI,
        filt="TESS"
    )
    fpp_liste[i] = ziel.FPP
    nfpp_liste[i] = ziel.NFPP
    print(f"  Durchlauf_{i+1}/{N_ITER}:  "
          f"FPP={fpp_liste[i]:.4f},  "
          f"NFPP={nfpp_liste[i]:.4f}")

# =====
# Schritt 8: Endergebnis
# =====
fpp_mittel = np.mean(fpp_liste)
fpp_std = np.std(fpp_liste)
nfpp_mittel = np.mean(nfpp_liste)
nfpp_std = np.std(nfpp_liste)

print(f"\nErgebnis (mit Contrast Curve, {N_ITER} Durchlaeufe):")
print(f"  FPP={fpp_mittel:.4f} +/- {fpp_std:.4f}")
print(f"  NFPP={nfpp_mittel:.4f} +/- {nfpp_std:.4f}")

# Validierungsstatus pruefen
if fpp_mittel < 0.015 and nfpp_mittel < 0.001:
    print("Status: VALIDIERTER PLANET")

elif fpp_mittel < 0.5:
    print("Status: PLANETENKANDIDAT (guenstig)")

else:
    print("Status: Nicht validiert")
    print("Grund: Flussverduennung durch helle Nachbarsterne"
          "in der TESS-Apertur (vgl. Diskussion)")

# Szenario-Tabelle ausgeben
print("\nSzenario-Wahrscheinlichkeiten:")
ergebnis = ziel.probs
for _, zeile in ergebnis.iterrows():
    print(f"  TIC_{int(zeile['ID']):>10}  "
          f"{zeile['scenario']:<8}  "
          f"P={zeile['prob']:.6f}")

```

## 14 Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle verwendeten Quellen sind vollständig und korrekt angegeben.

A handwritten signature in black ink, appearing to read 'Hindriks' followed by a stylized flourish and 'remini'.

Aarburg, 29.03.2026

---

Unterschrift des Autors